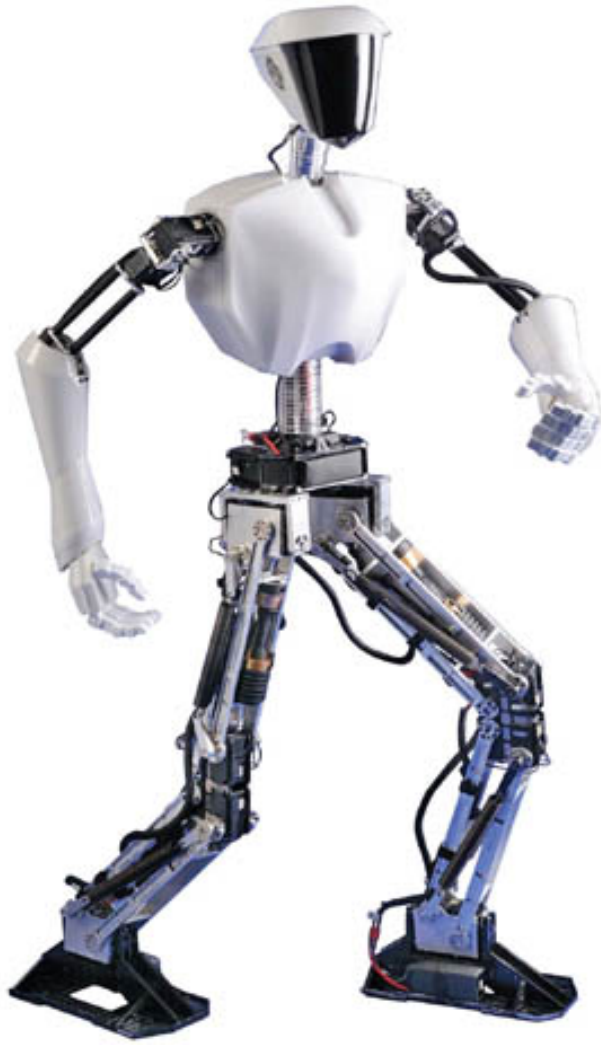
NORMAN S. NISE



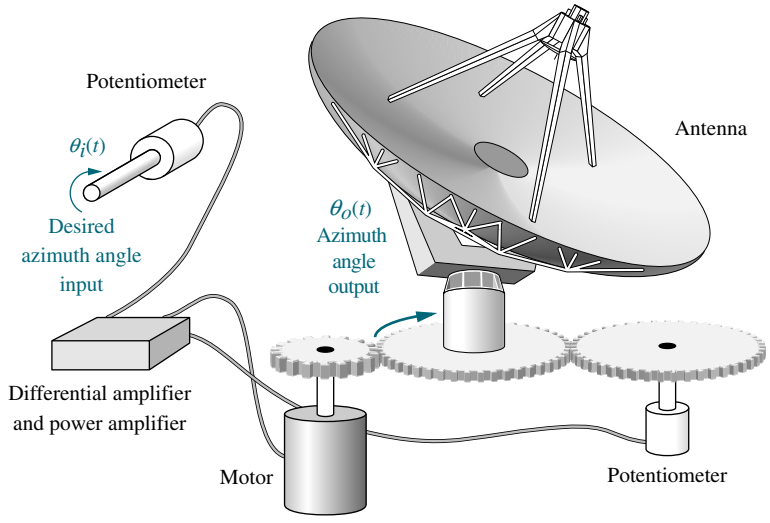
CONTROL SYSTEMS ENGINEERING

SIXTH EDITION

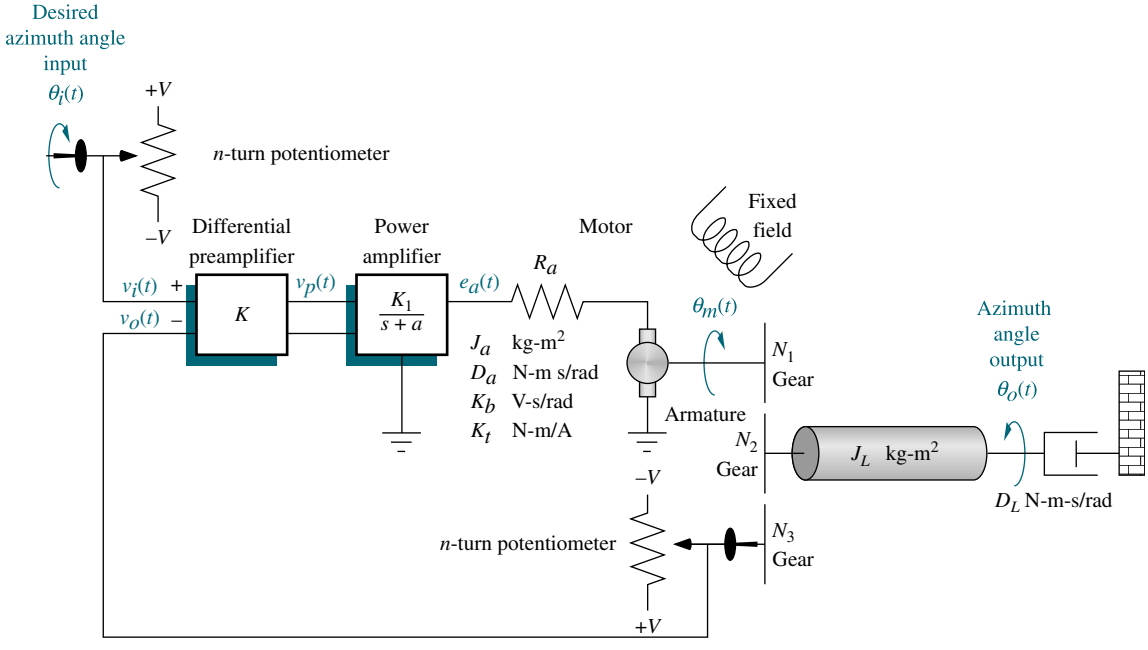
This page intentionally left blank

Antenna Azimuth Position Control System

Layout

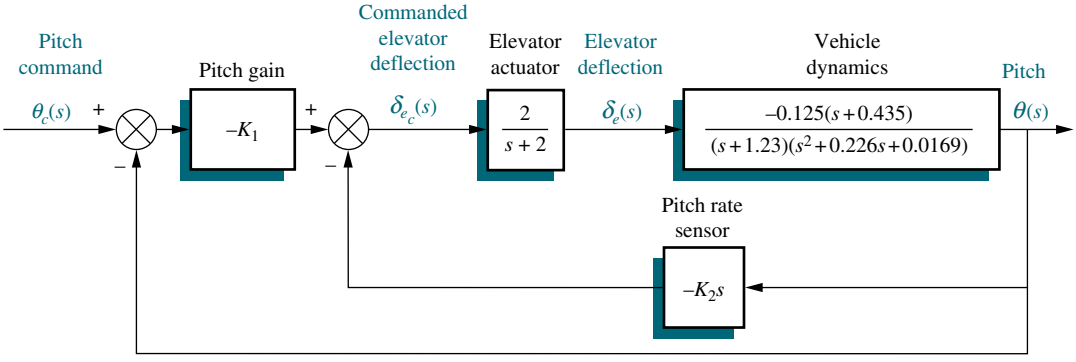


Schematic

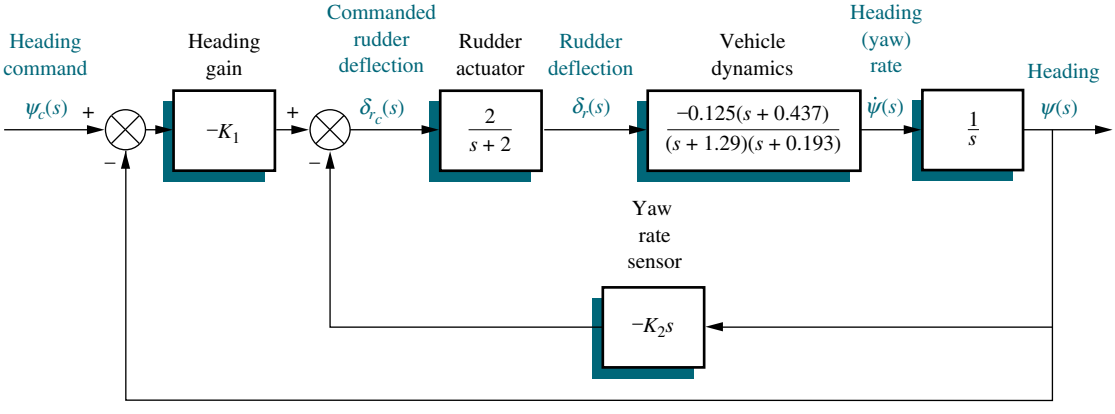


Unmanned Free-Swimming Submersible Vehicle

Pitch Control System



Heading Control System



WILEY PLUS

accessibly affordable
active learning

WileyPLUS is an innovative, research-based, online environment for effective teaching and learning.

WileyPLUS...

...motivates students with confidence-boosting feedback and proof of progress, 24/7.

...supports instructors with reliable resources that reinforce course goals inside and outside of the classroom.



Including Interactive Textbook & Resources

WileyPLUS... Learn More.

www.wileyplus.com

WILEY PLUS

ALL THE HELP, RESOURCES, AND PERSONAL SUPPORT YOU AND YOUR STUDENTS NEED

www.wileyplus.com/resources

1st DAY OF CLASS AND BEYOND

WILEY PLUS

WILEY FACULTY NETWORK

WILEY PLUS QuickStart



CONTROL SYSTEMS ENGINEERING
Sixth Edition

Norman S. Nise
California State Polytechnic University, Pomona



John Wiley & Sons, Inc.

To my wife, Ellen; sons, Benjamin and Alan; and daughter, Sharon, and their families.

Vice President & Publisher	Don Fowley
Publisher	Daniel Sayre
Senior Editorial Assistant	Katie Singleton
Associate Director of Marketing	Amy Scholz
Marketing Manager	Christopher Ruel
Production Manager	Dorothy Sinclair
Production Editor	Sandra Dumas
Creative Director	Harry Nolan
Cover Designer	James O'Shea
Cover Photo	Jim Stroup, Virginia Tech
Photo Department Manager	Hilary Newman
Photo Editor	Sheena Goldstein
Executive Media Editor	Thomas Kulesa
Associate Media Editor	Jennifer Mullin
Production Management Services	Integra Software Services Inc.

This book was typeset in 10/12 TimesRoman at Thomson and printed and bound by R. R. Donnelley (Jefferson City). The cover was printed by R. R. Donnelley (Jefferson City).

The paper in this book was manufactured by a mill whose forest management programs include sustained yield-harvesting of its timberlands. Sustained yield harvesting principles ensure that the number of trees cut each year does not exceed the amount of new growth.

This book is printed on acid-free paper. ♾

On the cover: CHARLI, a 5-foot tall autonomous humanoid robot built by Dr. Dennis Hong and his students at RoMeLa (Robotics and Mechanisms Laboratory) in the College of Engineering of Virginia Tech.

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work, in 2008, we launched a Corporate Citizenship initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: www.wiley.com/go/citizenship.

The software programs available with this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher and author do not offer any warranties or restrictions, nor do they accept any liabilities with respect to the programs.

Copyright © 2011, 2006, 2003, 1996 by John Wiley & Sons, Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774, (201) 748-6011, fax (201) 748-6008.

Evaluation copies are provided to qualified academics and professionals for review purposes only, for use in their courses during the next academic year. These copies are licensed and may not be sold or transferred to a third party. Upon completion of the review period, please return the evaluation copy to Wiley. Return instructions and a free of charge return shipping label are available at www.wiley.com/go/returnlabel. Outside of the United States, please contact your local representative.

ISBN 13 978-0470-54756-4
ISBN 13 978-0470-91769-5

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Contents

PREFACE, ix

1. INTRODUCTION, 1

1.1 Introduction, 2
1.2 A History of Control Systems, 4
1.3 System Configurations, 7
1.4 Analysis and Design Objectives, 10
Case Study, 12
1.5 The Design Process, 15
1.6 Computer-Aided Design, 20
1.7 The Control Systems Engineer, 21
Summary, 23
Review Questions, 23
Problems, 24
Cyber Exploration Laboratory, 30
Bibliography, 31

2. MODELING IN THE FREQUENCY DOMAIN, 33

2.1 Introduction, 34
2.2 Laplace Transform Review, 35
2.3 The Transfer Function, 44
2.4 Electrical Network Transfer Functions, 47
2.5 Translational Mechanical System Transfer Functions, 61
2.6 Rotational Mechanical System Transfer Functions, 69
2.7 Transfer Functions for Systems with Gears, 74
2.8 Electromechanical System Transfer Functions, 79
2.9 Electric Circuit Analogs, 84
2.10 Nonlinearities, 88
2.11 Linearization, 89
Case Studies, 94
Summary, 97
Review Questions, 97
Problems, 98
Cyber Exploration Laboratory, 112
Bibliography, 115

3. MODELING IN THE TIME DOMAIN, 117

3.1 Introduction, 118
3.2 Some Observations, 119
3.3 The General State-Space Representation, 123
3.4 Applying the State-Space Representation, 124
3.5 Converting a Transfer Function to State Space, 132
3.6 Converting from State Space to a Transfer Function, 139
3.7 Linearization, 141
Case Studies, 144
Summary, 148
Review Questions, 149
Problems, 149
Cyber Exploration Laboratory, 157
Bibliography, 159

4. TIME RESPONSE, 161

4.1 Introduction, 162
4.2 Poles, Zeros, and System Response, 162
4.3 First-Order Systems, 166
4.4 Second-Order Systems: Introduction, 168
4.5 The General Second-Order System, 173
4.6 Underdamped Second-Order Systems, 177
4.7 System Response with Additional Poles, 186
4.8 System Response With Zeros, 191
4.9 Effects of Nonlinearities Upon Time Response, 196

4.10 Laplace Transform Solution of State Equations, 199
4.11 Time Domain Solution of State Equations, 203
Case Studies, 207
Summary, 213
Review Questions, 214
Problems, 215
Cyber Exploration Laboratory, 228
Bibliography, 232

5. REDUCTION OF MULTIPLE SUBSYSTEMS, 235

5.1 Introduction, 236
5.2 Block Diagrams, 236
5.3 Analysis and Design of Feedback Systems, 245
5.4 Signal-Flow Graphs, 248
5.5 Mason's Rule, 251
5.6 Signal-Flow Graphs of State Equations, 254
5.7 Alternative Representations in State Space, 256
5.8 Similarity Transformations, 266
Case Studies, 272
Summary, 278
Review Questions, 279
Problems, 280
Cyber Exploration Laboratory, 297
Bibliography, 299

6. STABILITY, 301

6.1 Introduction, 302
6.2 Routh-Hurwitz Criterion, 305
6.3 Routh-Hurwitz Criterion: Special Cases, 308
6.4 Routh-Hurwitz Criterion: Additional Examples, 314
6.5 Stability in State Space, 320
Case Studies, 323
Summary, 325
Review Questions, 325
Problems, 326
Cyber Exploration Laboratory, 335
Bibliography, 336

7. STEADY-STATE ERRORS, 339

7.1 Introduction, 340
7.2 Steady-State Error for Unity Feedback Systems, 343
7.3 Static Error Constants and System Type, 349
7.4 Steady-State Error Specifications, 353
7.5 Steady-State Error for Disturbances, 356
7.6 Steady-State Error for Nonunity Feedback Systems, 358
7.7 Sensitivity, 362
7.8 Steady-State Error for Systems in State Space, 364
Case Studies, 368
Summary, 371
Review Questions, 372
Problems, 373
Cyber Exploration Laboratory, 384
Bibliography, 386

8. ROOT LOCUS TECHNIQUES, 387

8.1 Introduction, 388
8.2 Defining the Root Locus, 392
8.3 Properties of the Root Locus, 394
8.4 Sketching the Root Locus, 397
8.5 Refining the Sketch, 402
8.6 An Example, 411
8.7 Transient Response Design via Gain Adjustment, 415
8.8 Generalized Root Locus, 419
8.9 Root Locus for Positive-Feedback Systems, 421
8.10 Pole Sensitivity, 424
Case Studies, 426
Summary, 431
Review Questions, 432
Problems, 432
Cyber Exploration Laboratory, 450
Bibliography, 452

9. DESIGN VIA ROOT LOCUS, 455

9.1 Introduction, 456
9.2 Improving Steady-State Error via Cascade Compensation, 459
9.3 Improving Transient Response via Cascade Compensation, 469
9.4 Improving Steady-State Error and Transient Response, 482
9.5 Feedback Compensation, 495
9.6 Physical Realization of Compensation, 503
Case Studies, 508
Summary, 513
Review Questions, 514
Problems, 515
Cyber Exploration Laboratory, 530
Bibliography, 531

10. FREQUENCY RESPONSE TECHNIQUES, 533

10.1 Introduction, 534
10.2 Asymptotic Approximations: Bode Plots, 540
10.3 Introduction to the Nyquist Criterion, 559
10.4 Sketching the Nyquist Diagram, 564
10.5 Stability via the Nyquist Diagram, 569
10.6 Gain Margin and Phase Margin via the Nyquist Diagram, 574
10.7 Stability, Gain Margin, and Phase Margin via Bode Plots, 576
10.8 Relation Between Closed-Loop Transient and Closed-Loop Frequency Responses, 580
10.9 Relation Between Closed- and Open-Loop Frequency Responses, 583
10.10 Relation Between Closed-Loop Transient and Open-Loop Frequency Responses, 589
10.11 Steady-State Error Characteristics from Frequency Response, 593
10.12 Systems with Time Delay, 597
10.13 Obtaining Transfer Functions Experimentally, 602
Case Study, 606
Summary, 607
Review Questions, 609
Problems, 610
Cyber Exploration Laboratory, 621
Bibliography, 623

11. DESIGN VIA FREQUENCY RESPONSE, 625

11.1 Introduction, 626
11.2 Transient Response via Gain Adjustment, 627
11.3 Lag Compensation, 630
11.4 Lead Compensation, 635
11.5 Lag-Lead Compensation, 641
Case Studies, 650
Summary, 652
Review Questions, 653
Problems, 653
Cyber Exploration Laboratory, 660
Bibliography, 661

12. DESIGN VIA STATE SPACE, 663

12.1 Introduction, 664
12.2 Controller Design, 665
12.3 Controllability, 672
12.4 Alternative Approaches to Controller Design, 676
12.5 Observer Design, 682
12.6 Observability, 689
12.7 Alternative Approaches to Observer Design, 693
12.8 Steady-State Error Design Via Integral Control, 700
Case Study, 704
Summary, 709
Review Questions, 710
Problems, 711
Cyber Exploration Laboratory, 719
Bibliography, 721

13. DIGITAL CONTROL SYSTEMS, 723

13.1 Introduction , 724
13.2 Modeling the Digital Computer, 727

13.3 The *z*-Transform, 730
13.4 Transfer Functions, 735
13.5 Block Diagram Reduction, 739
13.6 Stability, 742
13.7 Steady-State Errors, 749
13.8 Transient Response on the *z*-Plane, 753
13.9 Gain Design on the *z*-Plane, 755
13.10 Cascade Compensation via the *s*-Plane, 758
13.11 Implementing the Digital Compensator, 762
Case Studies, 765
Summary, 769
Review Questions, 770
Problems, 771
Cyber Exploration Laboratory, 778
Bibliography, 780

Appendix A List of Symbols, 783

Appendix B MATLAB Tutorial, 787

B.1 Introduction, 787
B.2 MATLAB Examples, 788
B.3 Command Summary, 833
Bibliography, 835

Appendix C MATLAB's Simulink Tutorial, 836

C.1 Introduction, 836
C.2 Using Simulink, 836
C.3 Examples, 841
Summary, 855
Bibliography, 856

Appendix D LabVIEW Tutorial, 857

D.1 Introduction, 857
D.2 Control Systems Analysis, Design, and Simulation, 858
D.3 Using LabVIEW, 859
D.4 Analysis and Design Examples, 862
D.5 Simulation Examples, 876
Summary, 885
Bibliography, 886

Glossary, 887

Answers to Selected Problems, 897

Credits, 903

Index, 907

Appendix E MATLAB's GUI Tools Tutorial (Online)

Appendix F MATLAB's Symbolic Math Toolbox Tutorial (Online)

Appendix G Matrices, Determinants, and Systems of Equations (Online)

Appendix H Control System Computational Aids (Online)

Appendix I Derivation of a Schematic for a DC Motor (Online)

Appendix J Derivation of the Time Domain Solution of State Equations (Online)

Appendix K Solution of State Equations for $t_0 \neq 0$ (Online)

Appendix L Derivation of Similarity Transformations (Online)

Appendix M Root Locus Rules: Derivations (Online)

Control Systems Engineering Toolbox (Online)

Cyber Exploration Laboratory Experiments Covers Sheets (Online)

Lecture Graphics (Online)

Solutions to Skill-Assessment Exercises (Online)

Online location is www.wiley.com/college/nise

Preface

This book introduces students to the theory and practice of control systems engineering. The text emphasizes the practical application of the subject to the analysis and design of feedback systems.

The study of control systems engineering is essential for students pursuing degrees in electrical, mechanical, aerospace, biomedical, or chemical engineering. Control systems are found in a broad range of applications within these disciplines, from aircraft and spacecraft to robots and process control systems.

Control Systems Engineering is suitable for upper-division college and university engineering students and for those who wish to master the subject matter through self-study. The student using this text should have completed typical lower-division courses in physics and mathematics through differential equations. Other required background material, including Laplace transforms and linear algebra, is incorporated in the text, either within chapter discussions or separately in the appendixes or on the book's Companion Web site. This review material can be omitted without loss of continuity if the student does not require it.

Key Features

The key features of this sixth edition are:

- Standardized chapter organization
- Qualitative and quantitative explanations
- **Examples, Skill-Assessment Exercises,** and **Case Studies** throughout the text
- WileyPLUS content management system for students and professors
- **Cyber Exploration Laboratory and Virtual Experiments**
- Abundant illustrations
- Numerous end-of-chapter problems
- Emphasis on design
- Flexible coverage
- Emphasis on computer-aided analysis and design including MATLAB®[1] and LabVIEW®[2]

[1] MATLAB is a registered trademark of The MathWorks, Inc.

[2] LabVIEW is a registered trademark of National Instruments Corporation.

- Icons identifying major topics

Let us look at each feature in more detail.

Standardized Chapter Organization

Each chapter begins with a list of chapter learning outcomes, followed by a list of case study learning outcomes that relate to specific student performance in solving a practical case study problem, such as an antenna azimuth position control system.

Topics are then divided into clearly numbered and labeled sections containing explanations, examples, and, where appropriate, skill-assessment exercises with answers. These numbered sections are followed by one or more case studies, as will be outlined in a few paragraphs. Each chapter ends with a brief summary, several review questions requiring short answers, a set of homework problems, and experiments.

Qualitative and Quantitative Explanations

Explanations are clear and complete and, where appropriate, include a brief review of required background material. Topics build upon and support one another in a logical fashion. Groundwork for new concepts and terminology is carefully laid to avoid overwhelming the student and to facilitate self-study.

Although quantitative solutions are obviously important, a qualitative or intuitive understanding of problems and methods of solution is vital to producing the insight required to develop sound designs. Therefore, whenever possible, new concepts are discussed from a qualitative perspective before quantitative analysis and design are addressed. For example, in Chapter 8 the student can simply look at the root locus and describe qualitatively the changes in transient response that will occur as a system parameter, such as gain, is varied. This ability is developed with the help of a few simple equations from Chapter 4.

Examples, Skill-Assessment Exercises, and Case Studies

Explanations are clearly illustrated by means of numerous numbered and labeled **Examples** throughout the text. Where appropriate, sections conclude with **Skill-Assessment Exercises**. These are computation drills, most with answers that test comprehension and provide immediate feedback. Complete solutions can be found at www.wiley.com/college/nise.

Broader examples in the form of **Case Studies** can be found after the last numbered section of every chapter, with the exception of Chapter 1. These case studies arc practical application problems that demonstrate the concepts introduced in the chapter. Each case study concludes with a ‘‘Challenge’’ problem that students may work in order to test their understanding of the material.

One of the case studies, an antenna azimuth position control system, is carried throughout the book. The purpose is to illustrate the application of new material in each chapter to the same physical system, thus highlighting the continuity of the design process. Another, more challenging case study, involving

an Unmannered Free-Swimming Submersible Vehicle, is developed over the course of five chapters.

WileyPLUS Content Management System for Students and Professors

WileyPLUS
WPCS
Control Solutions

WileyPLUS is an online suite of resources, including the full text, for students and instructors. For the sixth edition of *Control Systems Engineering*, this suite offers professors who adopt the book with WileyPLUS the ability to create homework assignments based on algorithmic problems or multi-part questions, which guide the student through a problem. Instructors also have the capability to integrate assets, such as the simulations, into their lecture presentations. Students will find a Read, Study, and Practice zone to help them work through problems based on the ones offered in the text.

Control Solutions (prepared by JustAsk) are included in the WileyPLUS platform. The student will find simulations and Control Solutions in the Read, Study, and Practice zone. The Control Solutions are highlighted in the text with a WileyPLUS icon.

A new addition to the WileyPLUS platform for this edition are National Instruments and Quanser Virtual Laboratories. You will find references to them in sidebar entries throughout the textbook.

Visit www.wiley.com or contact your local Wiley representative for information.

Cyber Exploration Laboratory and Virtual Experiments

Computer experiments using MATLAB, Simulink®[3] and the Control System Toolbox are found at the end of the Problems sections under the sub-heading **Cyber Exploration Laboratory**. New to this edition is LabVIEW, which is also used for experiments within the Cyber Exploration Laboratory section of the chapters. The experiments allow the reader to verify the concepts covered in the chapter via simulation. The reader also can change parameters and perform "what if" exploration to gain insight into the effect of parameter and configuration changes. The experiments are written with stated Objectives, Minimum Required Software Packages, as well as Prelab, Lab, and Postlab tasks and questions. Thus, the experiments may be used for a laboratory course that accompanies the class. Cover sheets for these experiments are available at www.wiley.com.college/nise.

In addition, and new to this sixth edition, are **Virtual Experiments**. These experiments are more tightly focused than the **Cyber Exploration Laboratory** experiments and use LabVIEW and Quanser virtual hardware to illustrate immediate discussion and examples. The experiments are referenced in sidebars throughout some chapters.

[3] Simulink is a registered trademark of The MathWorks, Inc.

Abundant Illustrations

The ability to visualize concepts and processes is critical to the student's understanding. For this reason, approximately 800 photos, diagrams, graphs, and tables appear throughout the book to illustrate the topics under discussion.

Numerous End-of-Chapter Problems

Each chapter ends with a variety of homework problems that allow students to test their understanding of the material presented in the chapter. Problems vary in degree of difficulty and complexity, and most chapters include several practical, real-life problems to help maintain students' motivation. Also, the homework problems contain progressive analysis and design problems that use the same practical systems to demonstrate the concepts of each chapter.

Emphasis on Design



This textbook places a heavy emphasis on design. Chapters 8, 9, 11, 12 and 13 focus primarily on design. But. even in chapters that emphasize analysis, simple design examples are included wherever possible.

Throughout the book, design examples involving physical systems are identified by the icon shown in the margin. End-of-chapter problems that involve the design of physical systems are included under the separate heading **Design Problems,** and also in chapters covering design, under the heading **Progressive Analysis and Design Problems**. In these examples and problems, a desired response is specified, and the student must evaluate certain system parameters, such as gain, or specify a system configuration along with parameter values. In addition, the text includes numerous design examples and problems (not identified by an icon) that involve purely mathematical systems.

Because visualization is so vital to understanding design, this text carefully relates indirect design specifications to more familiar ones. For example, the less familiar and indirect phase margin is carefully related to the more direct and familiar percent overshoot before being used as a design specification.

For each general type of design problem introduced in the text, a methodology for solving the problem is presented—in many cases in the form of a step-by-step procedure, beginning with a statement of design objectives. Example problems serve to demonstrate the methodology by following the procedure, making simplifying assumptions, and presenting the results of the design in tables or plots that compare the performance of the original system to that of the improved system. This comparison also serves as a check on the simplifying assumptions.

Transient response design topics are covered comprehensively in the text. They include:

- Design via gain adjustment using the root locus
- Design of compensation and controllers via the root locus
- Design via gain adjustment using sinusoidal frequency response methods
- Design of compensation via sinusoidal frequency response methods

- Design of controllers in state space using pole-placement techniques
- Design of observers in state-space using pole-placement techniques
- Design of digital control systems via gain adjustment on the root locus
- Design of digital control system compensation via *s*-plane design and the Tustin transformation

Steady-state error design is covered comprehensively in this textbook and includes:

- Gain adjustment
- Design of compensation via the root locus
- Design of compensation via sinusoidal frequency response methods
- Design of integral control in state space

Finally, the design of gain to yield stability is covered from the following perspectives:

- Routh-Hurwitz criterion
- Root locus
- Nyquist criterion
- Bode plots

Flexible Coverage

The material in this book can be adapted for a one-quarter or a one-semester course. The organization is flexible, allowing the instructor to select the material that best suits the requirements and time constraints of the class.

State Space
SS

Throughout the book, state-space methods are presented along with the classical approach. Chapters and sections (as well as examples, exercises, review questions, and problems) that cover state space are marked by the icon shown in the margin and can be omitted without any loss of continuity. Those wishing to add a basic introduction to state-space modeling can include Chapter 3 in the syllabus.

In a one-semester course, the discussions of slate-space analysis in Chapters 4, 5, 6 and 7, as well as state-space design in Chapter 12, can be covered along with the classical approach. Another option is to teach state space separately by gathering the appropriate chapters and sections marked with the **State Space** icon into a single unit that follows the classical approach. In a one-quarter course, Chapter 13, "Digital Control Systems," could be eliminated.

Emphasis on Computer-Aided Analysis and Design

Control systems problems, particularly analysis and design problems using the root locus, can be tedious, since their solution involves trial and error. To solve these problems, students should be given access to computers or programmable calculators configured with appropriate software. In this sixth edition, MATLAB continues to be integrated into the text as an optional feature. In addition, and new to this

edition, we have included LabVIEW as an option to computer-aided analysis and design.

Many problems in this text can be solved with either a computer or a hand-held programmable calculator. For example, students can use the programmable calculator to (1) determine whether a point on the *s*-plane is also on the root locus, (2) find magnitude and phase frequency response data for Nyquist and Bode diagrams, and (3) convert between the following representations of a second-order system:

- Pole location in polar coordinates
- Pole location in Cartesian coordinates
- Characteristic polynomial
- Natural frequency and damping ratio
- Settling time and percent overshoot
- Peak time and percent overshoot
- Settling time and peak time

Handheld calculators have the advantage of easy accessibility for homework and exams. Please consult Appendix H, located at www.wiley.com/college/nise, for a discussion of computational aids that can be adapted to handheld calculators.

Personal computers are better suited for more computation-intensive applications, such as plotting time responses, root loci, and frequency response curves, as well as finding state-transition matrices. These computers also give the student a real-world environment in which to analyze and design control systems. Those not using MATLAB or LabVIEW can write their own programs or use other programs, such as Program CC. Please consult Appendix H at www.wiley.com/college/nise for a discussion of computational aids that can be adapted for use on computers that do not have MATLAB or LabVIEW installed.

Without access to computers or programmable calculators, students cannot obtain meaningful analysis and design results and the learning experience will be limited.

Icons Identifying Major Topics

WileyPLUS
WPCS
Control Solutions

MATLAB
ML

Simulink
SL

Gui Tool
GUIT

Several icons identify coverage and optional material. The icons are summarized as follows:

Control Solutions for the student are identified with a WileyPLUS icon. These problems, developed by JustAsk, are worked in detail and offer explanations of every facet of the solution.

The MATLAB icon identifies MATLAB discussions, examples, exercises, and problems. MATLAB coverage is provided as an enhancement and is not required to use the text.

The Simulink icon identifies Simulink discussions, examples, exercises, and problems. Simulink coverage is provided as an enhancement and is not required to use the text.

The GUI Tool icon identifies MATLAB GUI Tools discussions, examples, exercises, and problems. The discussion of the tools, which includes the LTI Viewer, the Simulink LTIViewer, and the SISO Design Tool, is provided as an enhancement and is not required to use the text.

Symbolic Math
SM

LabVIEW
LV

State Space
SS

Design
D

The Symbolic Math icon identifies Symbolic Math Toolbox discussions, examples, exercises, and problems. Symbolic Math Toolbox coverage is provided as an enhancement and is not required to use the text.

The LabVIEW icon identifies LabVIEW discussions, examples, exercises, and problems. LabVIEW is provided as an enhancement and is not required to use the text.

The State Space icon highlights state-space discussions, examples, exercises, and problems. State-space material is optional and can be omitted without loss of continuity.

The Design icon clearly identifies design problems involving physical systems.

New to This Edition

The following list describes the key changes in this sixth edition

End-of-chapter problems More than 20% of the end-of-chapter problems are either new or revised. Also, an additional Progressive Analysis and Design Problem has been added at the end of the chapter problems. The new progressive problem analyzes and designs a hybrid electric vehicle.

MATLAB The use of MATLAB for computer-aided analysis and design continues to be integrated into discussions and problems as an optional feature in the sixth edition. The MATLAB tutorial has been updated to MATLAB Version 7.9 (R 2009b), the Control System Toolbox Version 8.4, and the Symbolic Math Toolbox Version 5.3

In addition, MATLAB code continues to be incorporated in the chapters in the form of sidebar boxes entitled TryIt.

Virtual Experiments Virtual experiments, developed by National Instruments and Quanser, are included via sidebar references to experiments on WileyPLUS. The experiments are performed with 3-D simulations of Quanser hardware using developed LabVIEW VIs. Virtual Experiments are tightly focused and linked to a discussion or example.

Cyber Exploration Laboratory Experiments using LabVIEW have been added. Cyber Exploration Laboratory experiments are general in focus and are envisioned to be used in an associated lab class.

MATLAB's Simulink The use of Simulink to show the effects of nonlinearities upon the time response of open-loop and closed-loop systems appears again in this sixth edition. We also continue to use Simulink to demonstrate how to simulate digital systems. Finally, the Simulink tutorial has been updated to Simulink 7.4

Chapter 11 Lag-lead compensator design using Nichols charts has been added to Section 11.5.

LabVIEW New to this edition is LabVIEW. A tutorial for this tool is included in Appendix D. LabVIEW is used in Cyber Exploration Laboratory experiments and other problems throughout the textbook.

Book Companion Site (BCS) at www.wiley.com/college/nise

The BCS for the sixth edition includes various student and instructor resources. This free resource can be accessed by going to www.wiley.com/college/nise and clicking on Student Companion Site. Professors also access their password-protected resources on the Instructor Companion Site available through this url. Instructors should contact their Wiley sales representative for access.

For the Student:

- All M-files used in the MATLAB, Simulink, GUI Tools, and Symbolic Math Toolbox tutorials, as well as the TryIt exercises
- Copies of the Cyber Exploration Laboratory experiments for use as experiment cover sheets
- Solutions to the Skill-Assessment Exercises in the text
- LabVIEW Virtual Experiments and LabVIEW VIs used in Appendix D

For the Instructor;

- PowerPoint®[4] files containing the figures from the textbook
- Solutions to end-of-chapter problem sets
- Simulations, developed by JustAsk, for inclusion in lecture presentations

Book Organization by Chapter

Many times it is helpful lo understand an author's reasoning behind the organization of the course material. The following paragraphs hopefully shed light on this topic.

The primary goal of Chapter 1 is to motivate students. In this chapter, students learn about the many applications of control systems in everyday life and about the advantages of study and a career in this field. Control systems engineering design objectives, such as transient response, steady-state error, and stability, are introduced, as is the path to obtaining these objectives. New and unfamiliar terms also are included in the Glossary.

Many students have trouble with an early step in the analysis and design sequence: transforming a physical system into a schematic. This step requires many simplifying assumptions based on experience the typical college student does not yet possess. Identifying some of these assumptions in Chapter 1 helps to fill the experience gap.

Chapters 2, 3, and 5 address the representation of physical systems. Chapters 2 and 3 cover modeling of open-loop systems, using frequency response techniques and state-space techniques, respectively. Chapter 5 discusses the representation and reduction of systems formed of interconnected open-loop subsystems. Only a representative sample of physical systems can be covered in a textbook of this length. Electrical, mechanical (both translational and rotational), and electromechanical systems are used as examples of physical systems that are modeled, analyzed, and designed. Linearization of a nonlinear system—one technique used by the engineer to simplify a system in order to represent it mathematically—is also introduced.

Chapter 4 provides an introduction to system analysis, that is, finding and describing the output response of a system. It may seem more logical to reverse the order of Chapters 4 and 5, to present the material in Chapter 4 along with other chapters covering analysis. However, many years of teaching control systems have taught me that the sooner students see an application of the study of system representation, the higher their motivation levels remain.

Chapters 6, 7, 8, and 9 return to control systems analysis and design with the study of stability (Chapter 6), steady-state errors (Chapter 7), and transient response of higher-order systems using root locus techniques (Chapter 8). Chapter 9 covers design of compensators and controllers using the root locus.

[4] PowerPoint is a registered trademark of Microsoft Corporation.

Chapters 10 and 11 focus on sinusoidal frequency analysis and design. Chapter 10, like Chapter 8, covers basic concepts for stability, transient response, and steady-state-error analysis. However, Nyquist and Bode methods are used in place of root locus. Chapter 11, like Chapter 9, covers the design of compensators, but from the point of view of sinusoidal frequency techniques rather than root locus.

An introduction to state-space design and digital control systems analysis and design completes the text in Chapters 12 and 13, respectively. Although these chapters can be used as an introduction for students who will be continuing their study of control systems engineering, they are useful by themselves and as a supplement to the discussion of analysis and design in the previous chapters. The subject matter cannot be given a comprehensive treatment in two chapters, but the emphasis is clearly outlined and logically linked to the rest of the book.

Acknowledgments

The author would like to acknowledge the contributions of faculty and students, both at California State Polytechnic University, Pomona, and across the country, whose suggestions through all editions have made a positive impact on the new edition.

I am deeply indebted to my colleagues, Elhami T. Ibrahim, Salomon Oldak, and Norali Pernalete at California State Polytechnic University, Pomona for authoring the creative new problems you will find at the end of every chapter. Dr. Pernalete created the LabVIEW experiments and problems you will find in this new edition. The new progressive problem, hybrid vehicle, that is at the end of every chapter is the creation of Dr Ibrahim. In addition to his busy schedule as Electrical and Computer Engineering Department Chairman and author of many of the new problems, Professor Oldak also error checked new additions to the book and prevented glitches from ever reaching you, the reader.

I would like to express my appreciation to contributors to this sixth edition who participated in reviews, accuracy checking, surveys, or focus groups. They are: Jorge Aravena, Louisiana State University; Kurt Behpour, Cal Poly San Luis Obispo; Bill Diong, Texas Christian University; Sam Guccione, Eastern Illinois University; Pushkin Kachroo, Virginia Tech; Dmitriy Kalantarov, Cal State San Diego; Kamran Iqbal, University of Arkansas, Little Rock; Pushkin Kachroo, Virginia Tech; Kevin Lynch, Northwestern University; Tesfay Meressi, University of Massachusetts, Dartmouth; Luai Najim, University of Alabama at Birmingham; Dalton Nelson, University of Alabama at Birmingham; Marcio S. de Queiroz, Louisiana State University; John Ridgely, Cal Poly San Luis Obispo; John Schmitt, Oregon State University; Lili Tabrizi, California State University, Los Angeles; Raman Unnikrishnan, Cal State Fullerton; Stephen Williams, Milwaukee School of Engineering; Jiann-Shiou Yang, University of Minnesota, Duluth; and Ryan Zurakowski, University of Delaware.

The author would like to thank John Wiley & Sons, Inc. and its staff for once again providing professional support for this project through all phases of its development. Specifically, the following are due recognition for their contributions: Don Fowler, Vice President and Publisher, who gave full corporate support to the project; Daniel Sayre, Publisher, with whom I worked closely and who provided guidance and leadership throughout the development of the sixth edition; and Katie Singleton, Senior Editorial Assistant, who was always there to answer my questions and respond to my concerns in a professional manner. There are many others who

worked behind the scenes, but who should be thanked never the less. Rather than repeating their names and titles here, I refer the reader to the copyright page of this book where they are listed and credited. I am very thankful for their contributions.

Next, I want to acknowledge Integra Software Services, Inc. and its staff for turning the sixth edition manuscript into the finished product you are holding in your hands. Specifically, kudos go out to Heather Johnson, Managing Editor, who, once again, was always there to address my concerns in a timely and professional manner.

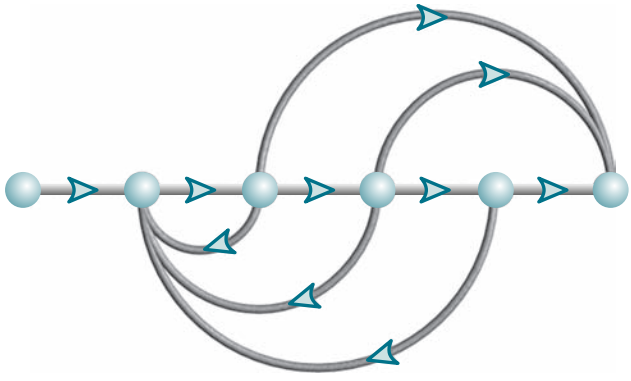
My sincere appreciation is extended to Erik Luther of National Instruments Corporation and Paul Gilbert and Michel Levis of Quanser for conceiving, coordinating, and developing the Virtual Experiments that I am sure will enhance your understanding of control systems.

Finally, last but certainly not least, I want to express my appreciation to my wife, Ellen, for her support in ways too numerous to mention during the writing of the past six editions. Specifically though, thanks to her proofing final pages for this sixth edition, you the reader hopefully will find comprehension rather than apprehension in the pages that follow.

Norman S. Nise

Introduction

1



Chapter Learning Outcomes

After completing this chapter, the student will be able to:

- Define a control system and describe some applications (Section 1.1)
- Describe historical developments leading to modern day control theory (Section 1.2)
- Describe the basic features and configurations of control systems (Section 1.3)
- Describe control systems analysis and design objectives (Section 1.4)
- Describe a control system's design process (Sections 1.5–1.6)
- Describe the benefit from studying control systems (Section 1.7)

Case Study Learning Outcomes

- You will be introduced to a running case study—an antenna azimuth position control system—that will serve to illustrate the principles in each subsequent chapter. In this chapter, the system is used to demonstrate qualitatively how a control system works as well as to define performance criteria that are the basis for control systems analysis and design.

1.1 Introduction

Control systems are an integral part of modern society. Numerous applications are all around us: The rockets fire, and the space shuttle lifts off to earth orbit; in splashing cooling water, a metallic part is automatically machined; a self-guided vehicle delivering material to workstations in an aerospace assembly plant glides along the floor seeking its destination. These are just a few examples of the automatically controlled systems that we can create.

We are not the only creators of automatically controlled systems; these systems also exist in nature. Within our own bodies are numerous control systems, such as the pancreas, which regulates our blood sugar. In time of "fight or flight," our adrenaline increases along with our heart rate, causing more oxygen to be delivered to our cells. Our eyes follow a moving object to keep it in view; our hands grasp the object and place it precisely at a predetermined location.

Even the nonphysical world appears to be automatically regulated. Models have been suggested showing automatic control of student performance. The input to the model is the student's available study time, and the output is the grade. The model can be used to predict the time required for the grade to rise if a sudden increase in study time is available. Using this model, you can determine whether increased study is worth the effort during the last week of the term.

Control System Definition

A control system consists of *subsystems* and *processes* (or *plants)* assembled for the purpose of obtaining a desired *output* with desired *performance*, given a specified *input*. Figure 1.1 shows a control system in its simplest form, where the input represents a desired output.

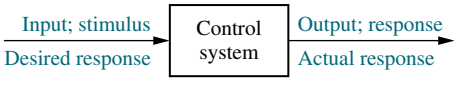


FIGURE 1.1 Simplified description of a control system

For example, consider an elevator. When the fourth-floor button is pressed on the first floor, the elevator rises to the fourth floor with a speed and floor-leveling accuracy designed for passenger comfort. The push of the fourth-floor button is an *input* that represents our desired *output*, shown as a step function in Figure 1.2. The *performance* of the elevator can be seen from the elevator response curve in the figure.

Two major measures of performance are apparent: (1) the transient response and (2) the steady-state error. In our example, passenger comfort and passenger patience are dependent upon the transient response. If this response is too fast, passenger comfort is sacrificed; if too slow, passenger patience is sacrificed. The steady-state error is another important performance specification since passenger safety and convenience would be sacrificed if the elevator did not properly level.

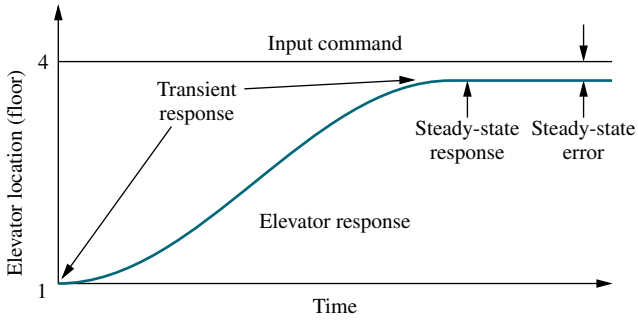


FIGURE 1.2 Elevator response



(a)



(b)

FIGURE 1.3 **a.** Early elevators were controlled by hand ropes or an elevator operator. Here a rope is cut to demonstrate the safety brake, an innovation in early elevators (© Bettman/Corbis); **b.** One of two modern Duo-lift elevators makes its way up the Grande Arche in Paris. Two elevators are driven by one motor, with each car acting as a counterbalance to the other. Today, elevators are fully automatic, using control systems to regulate position and velocity.

Advantages of Control Systems

With control systems we can move large equipment with precision that would otherwise be impossible. We can point huge antennas toward the farthest reaches of the universe to pick up faint radio signals; controlling these antennas by hand would be impossible. Because of control systems, elevators carry us quickly to our destination, automatically stopping at the right floor (Figure 1.3). We alone could not provide the power required for the load and the speed; motors provide the power, and control systems regulate the position and speed.

We build control systems for four primary reasons:

1. Power amplification
2. Remote control
3. Convenience of input form
4. Compensation for disturbances

For example, a radar antenna, positioned by the low-power rotation of a knob at the input, requires a large amount of power for its output rotation. A control system can produce the needed power amplification, or power *gain*.

Robots designed by control system principles can compensate for human disabilities. Control systems are also useful in remote or dangerous locations. For example, a remote-controlled robot arm can be used to pick up material in a radioactive environment. Figure 1.4 shows a robot arm designed to work in contaminated environments.

Control systems can also be used to provide convenience by changing the form of the input. For example, in a temperature control system, the input is a *position* on a thermostat. The output is *heat*. Thus, a convenient position input yields a desired thermal output.

Another advantage of a control system is the ability to compensate for disturbances. Typically, we control such variables as temperature in

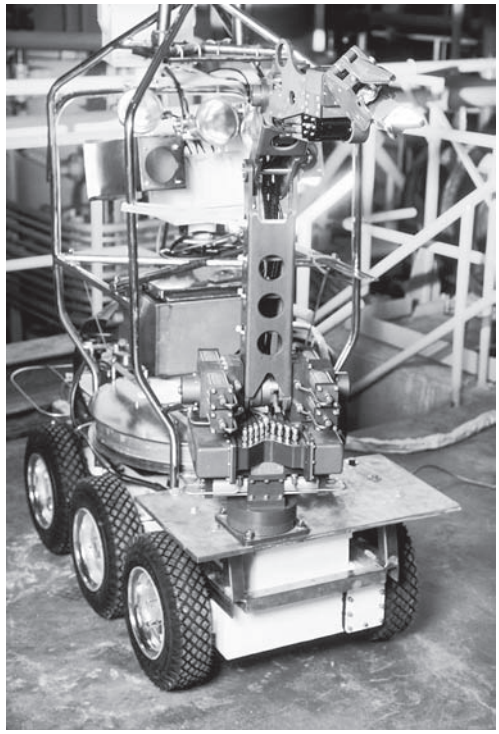


FIGURE 1.4 *Rover* was built to work in contaminated areas at Three Mile Island in Middleton, Pennsylvania, where a nuclear accident occurred in 1979. The remote-controlled robot's long arm can be seen at the front of the vehicle.

thermal systems, position and velocity in mechanical systems, and voltage, current, or frequency in electrical systems. The system must be able to yield the correct output even with a disturbance. For example, consider an antenna system that points in a commanded direction. If wind forces the antenna from its commanded position, or if noise enters internally, the system must be able to detect the disturbance and correct the antenna's position. Obviously, the system's input will not change to make the correction. Consequently, the system itself must measure the amount that the disturbance has repositioned the antenna and then return the antenna to the position commanded by the input.

1.2 A History of Control Systems

Feedback control systems are older than humanity. Numerous biological control systems were built into the earliest inhabitants of our planet. Let us now look at a brief history of human-designed control systems.[1]

Liquid-Level Control

The Greeks began engineering feedback systems around 300 B.C. A water clock invented by Ktesibios operated by having water trickle into a measuring container at a constant rate. The level of water in the measuring container could be used to tell time. For water to trickle at a constant rate, the supply tank had to be kept at a constant level. This was accomplished using a float valve similar to the water-level control in today's flush toilets.

Soon after Ktesibios, the idea of liquid-level control was applied to an oil lamp by Philon of Byzantium. The lamp consisted of two oil containers configured vertically. The lower pan was open at the top and was the fuel supply for the flame. The closed upper bowl was the fuel reservoir for the pan below. The containers were interconnected by two capillary tubes and another tube, called a *vertical riser*, which was inserted into the oil in the lower pan just below the surface. As the oil burned, the base of the vertical riser was exposed to air, which forced oil in the reservoir above to flow through the capillary tubes and into the pan. The transfer of fuel from the upper reservoir to the pan stopped when the previous oil level in the pan was reestablished, thus blocking the air from entering the vertical riser. Hence, the system kept the liquid level in the lower container constant.

Steam Pressure and Temperature Controls

Regulation of steam pressure began around 1681 with Denis Papin's invention of the safety valve. The concept was further elaborated on by weighting the valve top. If the upward pressure from the boiler exceeded the weight, steam was released, and the pressure decreased. If it did not exceed the weight, the valve did not open, and the pressure inside the boiler increased. Thus, the weight on the valve top set the internal pressure of the boiler.

Also in the seventeenth century, Cornelis Drebbel in Holland invented a purely mechanical temperature control system for hatching eggs. The device used a vial of alcohol and mercury with a floater inserted in it. The floater was connected to a damper that controlled a flame. A portion of the vial was inserted into the incubator to sense the heat generated by the fire. As the heat increased, the alcohol and mercury expanded, raising the floater, closing the damper, and reducing the flame. Lower temperature caused the float to descend, opening the damper and increasing the flame.

Speed Control

In 1745, speed control was applied to a windmill by Edmund Lee. Increasing winds pitched the blades farther back, so that less area was available. As the wind

[1] See *Bennett* (*1979*) and *Mayr* (*1970*) for definitive works on the history of control systems.

decreased, more blade area was available. William Cubitt improved on the idea in 1809 by dividing the windmill sail into movable louvers.

Also in the eighteenth century, James Watt invented the flyball speed governor to control the speed of steam engines. In this device, two spinning flyballs rise as rotational speed increases. A steam valve connected to the flyball mechanism closes with the ascending flyballs and opens with the descending flyballs, thus regulating the speed.

Stability, Stabilization, and Steering

Control systems theory as we know it today began to crystallize in the latter half of the nineteenth century. In 1868, James Clerk Maxwell published the stability criterion for a third-order system based on the coefficients of the differential equation. In 1874, Edward John Routh, using a suggestion from William Kingdon Clifford that was ignored earlier by Maxwell, was able to extend the stability criterion to fifth-order systems. In 1877, the topic for the Adams Prize was "The Criterion of Dynamical Stability." In response, Routh submitted a paper entitled *A Treatise on the Stability of a Given State of Motion* and won the prize. This paper contains what is now known as the Routh-Hurwitz criterion for stability, which we will study in Chapter 6. Alexandr Michailovich Lyapunov also contributed to the development and formulation of today's theories and practice of control system stability. A student of P. L. Chebyshev at the University of St. Petersburg in Russia, Lyapunov extended the work of Routh to nonlinear systems in his 1892 doctoral thesis, entitled *The General Problem of Stability of Motion*.

During the second half of the 1800s, the development of control systems focused on the steering and stabilizing of ships. In 1874, Henry Bessemer, using a gyro to sense a ship's motion and applying power generated by the ship's hydraulic system, moved the ship's saloon to keep it stable (whether this made a difference to the patrons is doubtful). Other efforts were made to stabilize platforms for guns as well as to stabilize entire ships, using pendulums to sense the motion.

Twentieth-Century Developments

It was not until the early 1900s that automatic steering of ships was achieved. In 1922, the Sperry Gyroscope Company installed an automatic steering system that used the elements of compensation and adaptive control to improve performance. However, much of the general theory used today to improve the performance of automatic control systems is attributed to Nicholas Minorsky, a Russian born in 1885. It was his theoretical development applied to the automatic steering of ships that led to what we call today proportional-plus-integral-plus-derivative (PID), or three-mode, controllers, which we will study in Chapters 9 and 11.

In the late 1920s and early 1930s, H. W. Bode and H. Nyquist at Bell Telephone Laboratories developed the analysis of feedback amplifiers. These contributions evolved into sinusoidal frequency analysis and design techniques currently used for feedback control system, and are presented in Chapters 10 and 11.

In 1948, Walter R. Evans, working in the aircraft industry, developed a graphical technique to plot the roots of a characteristic equation of a feedback system whose parameters changed over a particular range of values. This technique, now known as the root locus, takes its place with the work of Bode and Nyquist in forming the foundation of linear control systems analysis and design theory. We will study root locus in Chapters 8, 9, and 13.

Contemporary Applications

Today, control systems find widespread application in the guidance, navigation, and control of missiles and spacecraft, as well as planes and ships at sea. For example,

modern ships use a combination of electrical, mechanical, and hydraulic components to develop rudder commands in response to desired heading commands. The rudder commands, in turn, result in a rudder angle that steers the ship.

We find control systems throughout the process control industry, regulating liquid levels in tanks, chemical concentrations in vats, as well as the thickness of fabricated material. For example, consider a thickness control system for a steel plate finishing mill. Steel enters the finishing mill and passes through rollers. In the finishing mill, X-rays measure the actual thickness and compare it to the desired thickness. Any difference is adjusted by a screw-down position control that changes the roll gap at the rollers through which the steel passes. This change in roll gap regulates the thickness.

Modern developments have seen widespread use of the digital computer as part of control systems. For example, computers in control systems are for industrial robots, spacecraft, and the process control industry. It is hard to visualize a modern control system that does not use a digital computer.

The space shuttle contains numerous control systems operated by an onboard computer on a time-shared basis. Without control systems, it would be impossible to guide the shuttle to and from earth's orbit or to adjust the orbit itself and support life on board. Navigation functions programmed into the shuttle's computers use data from the shuttle's hardware to estimate vehicle position and velocity. This information is fed to the guidance equations that calculate commands for the shuttle's flight control systems, which steer the spacecraft. In space, the flight control system gimbals (rotates) the orbital maneuvering system (OMS) engines into a position that provides thrust in the commanded direction to steer the spacecraft. Within the earth's atmosphere, the shuttle is steered by commands sent from the flight control system to the aerosurfaces, such as the elevons.

Within this large control system represented by navigation, guidance, and control are numerous subsystems to control the vehicle's functions. For example, the elevons require a control system to ensure that their position is indeed that which was commanded, since disturbances such as wind could rotate the elevons away from the commanded position. Similarly, in space, the gimbaling of the orbital maneuvering engines requires a similar control system to ensure that the rotating engine can accomplish its function with speed and accuracy. Control systems are also used to control and stabilize the vehicle during its descent from orbit. Numerous small jets that compose the reaction control system (RCS) are used initially in the exoatmosphere, where the aerosurfaces are ineffective. Control is passed to the aerosurfaces as the orbiter descends into the atmosphere.

Inside the shuttle, numerous control systems are required for power and life support. For example, the orbiter has three fuel-cell power plants that convert hydrogen and oxygen (reactants) into electricity and water for use by the crew. The fuel cells involve the use of control systems to regulate temperature and pressure. The reactant tanks are kept at constant pressure as the quantity of reactant diminishes. Sensors in the tanks send signals to the control systems to turn heaters on or off to keep the tank pressure constant (*Rockwell International*, *1984*).

Control systems are not limited to science and industry. For example, a home heating system is a simple control system consisting of a thermostat containing a bimetallic material that expands or contracts with changing temperature. This expansion or contraction moves a vial of mercury that acts as a switch, turning the heater on or off. The amount of expansion or contraction required to move the mercury switch is determined by the temperature setting.

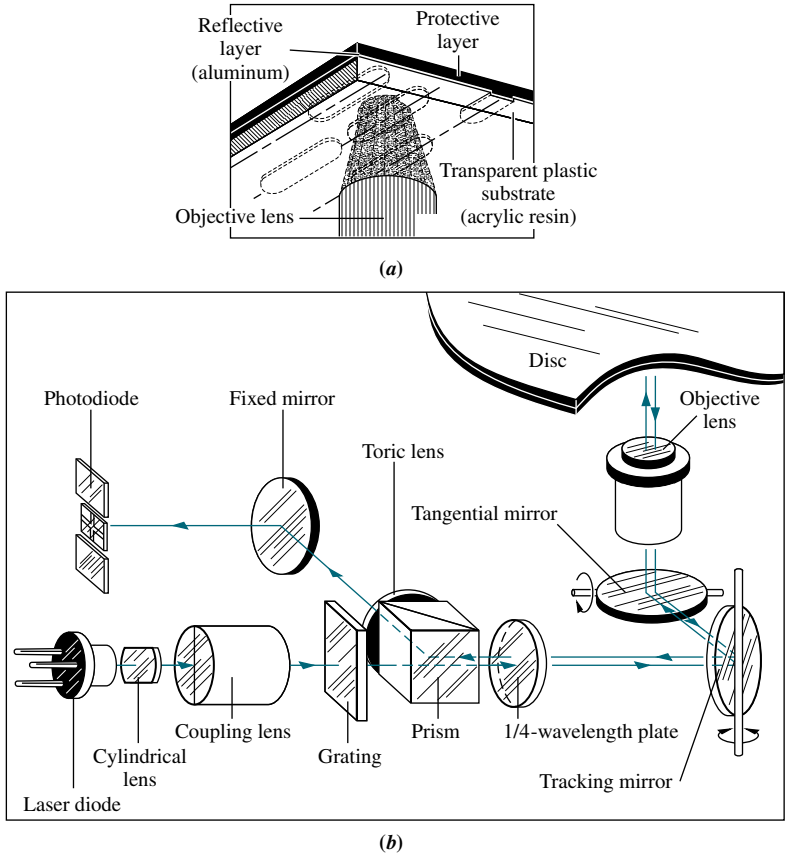


FIGURE 1.5 Optical playback system: **a.** objective lens reading pits on an optical disc; **b.** optical path for playback, showing tracking mirror rotated by a control system to keep the laser beam positioned on the pits (Pioneer Electronics (USA), Inc.)

Home entertainment systems also have built-in control systems. For example, in an optical disk recording system microscopic pits representing the information are burned into the disc by a laser during the recording process. During playback, a reflected laser beam focused on the pits changes intensity (Figure 1.5). The light intensity changes are converted to an electrical signal and processed as sound or picture. A control system keeps the laser beam positioned on the pits, which are cut as concentric circles.

There are countless other examples of control systems, from the everyday to the extraordinary. As you begin your study of control systems engineering, you will become more aware of the wide variety of applications.

1.3 System Configurations

In this section, we discuss two major configurations of control systems: open loop and closed loop. We can consider these configurations to be the internal architecture of the total system shown in Figure 1.1. Finally, we show how a digital computer forms part of a control system's configuration.

Open-Loop Systems

A generic *open-loop system* is shown in Figure 1.6(*a*). It starts with a subsystem called an *input transducer*, which converts the form of the input to that used by the *controller*. The controller drives a *process* or a *plant*. The input is sometimes called the *reference*, while the output can be called the *controlled variable*. Other signals, such as *disturbances*, are shown added to the controller and process outputs via *summing junctions*, which yield the algebraic sum of their input signals using associated signs. For example, the plant can be a furnace or air conditioning system, where the output variable is temperature. The controller in a heating system consists of fuel valves and the electrical system that operates the valves.

The distinguishing characteristic of an open-loop system is that it cannot compensate for any disturbances that add to the controller's driving signal (Disturbance 1 in Figure 1.6(*a*)). For example, if the controller is an electronic amplifier and Disturbance 1 is noise, then any additive amplifier noise at the first summing junction will also drive the process, corrupting the output with the effect of the noise. The output of an open-loop system is corrupted not only by signals that add to the controller's commands but also by disturbances at the output (Disturbance 2 in Figure 1.6(*a*)). The system cannot correct for these disturbances, either.

Open-loop systems, then, do not correct for disturbances and are simply commanded by the input. For example, toasters are open-loop systems, as anyone with burnt toast can attest. The controlled variable (output) of a toaster is the color of the toast. The device is designed with the assumption that the toast will be darker the longer it is subjected to heat. The toaster does not measure the color of the toast; it does not correct for the fact that the toast is rye, white, or sourdough, nor does it correct for the fact that toast comes in different thicknesses.

Other examples of open-loop systems are mechanical systems consisting of a mass, spring, and damper with a constant force positioning the mass. The greater the force, the greater the displacement. Again, the system position will change with a disturbance, such as an additional force, and the system will not detect or correct for the disturbance. Or assume that you calculate the amount of time you need to study

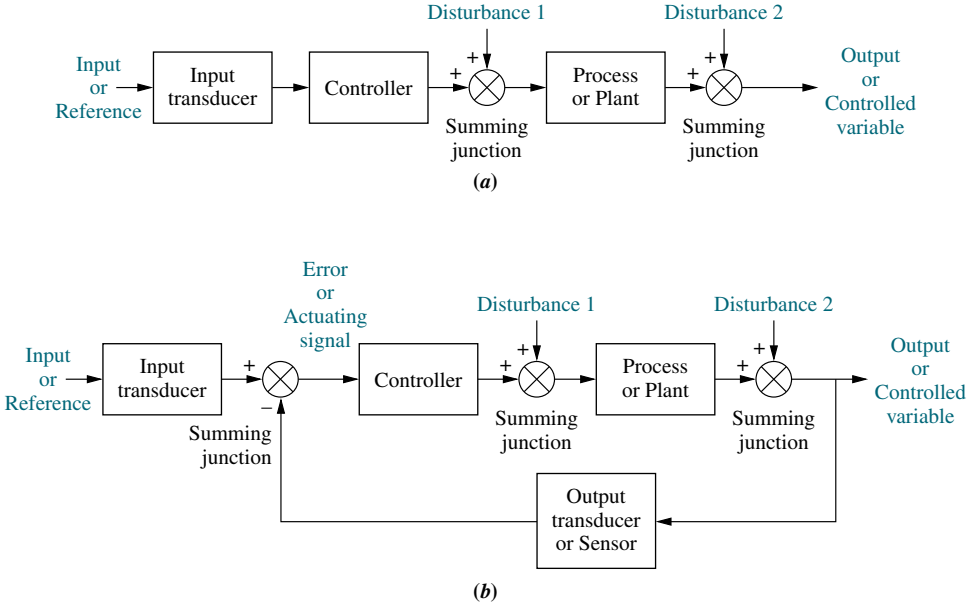


FIGURE 1.6 Block diagrams of control systems: **a.** open-loop system; **b.** closed-loop system

for an examination that covers three chapters in order to get an A. If the professor adds a fourth chapter—a disturbance—you are an open-loop system if you do not detect the disturbance and add study time to that previously calculated. The result of this oversight would be a lower grade than you expected.

Closed-Loop (Feedback Control) Systems

The disadvantages of open-loop systems, namely sensitivity to disturbances and inability to correct for these disturbances, may be overcome in *closed-loop systems*. The generic architecture of a closed-loop system is shown in Figure 1.6(*b*).

The input transducer converts the form of the input to the form used by the controller. An *output transducer*, or *sensor*, measures the output response and converts it into the form used by the controller. For example, if the controller uses electrical signals to operate the valves of a temperature control system, the input position and the output temperature are converted to electrical signals. The input position can be converted to a voltage by a *potentiometer*, a variable resistor, and the output temperature can be converted to a voltage by a *thermistor*, a device whose electrical resistance changes with temperature.

The first summing junction algebraically adds the signal from the input to the signal from the output, which arrives via the *feedback path*, the return path from the output to the summing junction. In Figure 1.6(*b*), the output signal is subtracted from the input signal. The result is generally called the *actuating signal*. However, in systems where both the input and output transducers have *unity gain* (that is, the transducer amplifies its input by 1), the actuating signal's value is equal to the actual difference between the input and the output. Under this condition, the actuating signal is called the *error*.

The closed-loop system compensates for disturbances by measuring the output response, feeding that measurement back through a feedback path, and comparing that response to the input at the summing junction. If there is any difference between the two responses, the system drives the plant, via the actuating signal, to make a correction. If there is no difference, the system does not drive the plant, since the plant's response is already the desired response.

Closed-loop systems, then, have the obvious advantage of greater accuracy than open-loop systems. They are less sensitive to noise, disturbances, and changes in the environment. Transient response and steady-state error can be controlled more conveniently and with greater flexibility in closed-loop systems, often by a simple adjustment of gain (amplification) in the loop and sometimes by redesigning the controller. We refer to the redesign as *compensating* the system and to the resulting hardware as a *compensator*. On the other hand, closed-loop systems are more complex and expensive than open-loop systems. A standard, open-loop toaster serves as an example: It is simple and inexpensive. A closed-loop toaster oven is more complex and more expensive since it has to measure both color (through light reflectivity) and humidity inside the toaster oven. Thus, the control systems engineer must consider the trade-off between the simplicity and low cost of an open-loop system and the accuracy and higher cost of a closed-loop system.

In summary, systems that perform the previously described measurement and correction are called closed-loop, or feedback control, systems. Systems that do not have this property of measurement and correction are called open-loop systems.

Computer-Controlled Systems

In many modern systems, the controller (or compensator) is a digital computer. The advantage of using a computer is that many loops can be controlled or compensated by the same computer through time sharing. Furthermore, any adjustments of the

compensator parameters required to yield a desired response can be made by changes in software rather than hardware. The computer can also perform supervisory functions, such as scheduling many required applications. For example, the space shuttle main engine (SSME) controller, which contains two digital computers, alone controls numerous engine functions. It monitors engine sensors that provide pressures, temperatures, flow rates, turbopump speed, valve positions, and engine servo valve actuator positions. The controller further provides closed-loop control of thrust and propellant mixture ratio, sensor excitation, valve actuators, spark igniters, as well as other functions (*Rockwell International*, *1984*).

1.4 Analysis and Design Objectives

In Section 1.1 we briefly alluded to some control system performance specifications, such as transient response and steady-state error. We now expand upon the topic of performance and place it in perspective as we define our analysis and design objectives.

Analysis is the process by which a system's performance is determined. For example, we evaluate its transient response and steady-state error to determine if they meet the desired specifications. *Design* is the process by which a system's performance is created or changed. For example, if a system's transient response and steady-state error are analyzed and found not to meet the specifications, then we change parameters or add additional components to meet the specifications.

A control system is *dynamic:* It responds to an input by undergoing a transient response before reaching a steady-state response that generally resembles the input. We have already identified these two responses and cited a position control system (an elevator) as an example. In this section, we discuss three major objectives of systems analysis and design: producing the desired transient response, reducing steady-state error, and achieving stability. We also address some other design concerns, such as cost and the sensitivity of system performance to changes in parameters.

Transient Response

Transient response is important. In the case of an elevator, a slow transient response makes passengers impatient, whereas an excessively rapid response makes them uncomfortable. If the elevator oscillates about the arrival floor for more than a second, a disconcerting feeling can result. Transient response is also important for structural reasons: Too fast a transient response could cause permanent physical damage. In a computer, transient response contributes to the time required to read from or write to the computer's disk storage (see Figure 1.7). Since reading and writing cannot take place until the head stops, the speed of the read/write head's movement from one track on the disk to another influences the overall speed of the computer.



FIGURE 1.7 Computer hard disk drive, showing disks and read/write head

In this book, we establish quantitative definitions for transient response. We then analyze the system for its *existing* transient response. Finally, we adjust parameters or design components to yield a *desired* transient response—our first analysis and design objective.

Steady-State Response

Another analysis and design goal focuses on the steady-state response. As we have seen, this response resembles the input and is usually what remains after the transients have decayed to zero. For example, this response may be an elevator stopped near the fourth floor or the head of a disk drive finally stopped at the correct track. We are concerned about the accuracy of the steady-state response. An elevator must be level enough with the floor for the passengers to exit, and a read/write head not positioned over the commanded track results in computer errors. An antenna tracking a satellite must keep the satellite well within its beamwidth in order not to lose track. In this text we define steady-state errors quantitatively, analyze a system's steady-state error, and then design corrective action to reduce the steady-state error—our second analysis and design objective.

Stability

Discussion of transient response and steady-state error is moot if the system does not have *stability*. In order to explain stability, we start from the fact that the total response of a system is the sum of the *natural response* and the *forced response*. When you studied linear differential equations, you probably referred to these responses as the *homogeneous* and the *particular solutions*, respectively. Natural response describes the way the system dissipates or acquires energy. The form or nature of this response is dependent only on the system, not the input. On the other hand, the form or nature of the forced response is dependent on the input. Thus, for a *linear* system, we can write

$$\text{Total response} = \text{Natural response} + \text{Forced response} \qquad (1.1)^2$$

For a control system to be useful, the natural response must (1) eventually approach zero, thus leaving only the forced response, or (2) oscillate. In some systems, however, the natural response grows without bound rather than diminish to zero or oscillate. Eventually, the natural response is so much greater than the forced response that the system is no longer controlled. This condition, called *instability*, could lead to self-destruction of the physical device if limit stops are not part of the design. For example, the elevator would crash through the floor or exit through the ceiling; an aircraft would go into an uncontrollable roll; or an antenna commanded to point to a target would rotate, line up with the target, but then begin to oscillate about the target with *growing* oscillations and *increasing* velocity until the motor or amplifiers reached their output limits or until the antenna was damaged structurally. A time plot of an unstable system would show a transient response that grows without bound and without any evidence of a steady-state response.

Control systems must be designed to be stable. That is, their natural response must decay to zero as time approaches infinity, or oscillate. In many systems the transient response you see on a time response plot can be directly related to the natural response. Thus, if the natural response decays to zero as time approaches infinity, the transient response will also die out, leaving only the forced response. If the system is stable, the proper transient response and steady-state error characteristics can be designed. Stability is our third analysis and design objective.

---

[2] You may be confused by the words *transient* vs. *natural*, and *steady-state* vs. *forced*. If you look at Figure 1.2, you can see the transient and steady-state portions of the total response as indicated. The transient response is the sum of the natural and forced responses, while the natural response is large. If we plotted the natural response by itself, we would get a curve that is different from the transient portion of Figure 1.2. The steady-state response of Figure 1.2 is also the sum of the natural and forced responses, but the natural response is small. Thus, the transient and steady-state responses are what you actually see on the plot; the natural and forced responses are the underlying mathematical components of those responses.

Other Considerations

The three main objectives of control system analysis and design have already been enumerated. However, other important considerations must be taken into account. For example, factors affecting hardware selection, such as motor sizing to fulfill power requirements and choice of sensors for accuracy, must be considered early in the design.

Finances are another consideration. Control system designers cannot create designs without considering their economic impact. Such considerations as budget allocations and competitive pricing must guide the engineer. For example, if your product is one of a kind, you may be able to create a design that uses more expensive components without appreciably increasing total cost. However, if your design will be used for many copies, slight increases in cost per copy can translate into many more dollars for your company to propose during contract bidding and to outlay before sales.

Another consideration is *robust* design. System parameters considered constant during the design for transient response, steady-state errors, and stability change over time when the actual system is built. Thus, the performance of the system also changes over time and will not be consistent with your design. Unfortunately, the relationship between parameter changes and their effect on performance is not linear. In some cases, even in the same system, changes in parameter values can lead to small or large changes in performance, depending on the system's nominal operating point and the type of design used. Thus, the engineer wants to create a robust design so that the system will not be sensitive to parameter changes. We discuss the concept of system sensitivity to parameter changes in Chapters 7 and 8. This concept, then, can be used to test a design for robustness.

Case Study

Introduction to a Case Study



FIGURE 1.8 The search for extraterrestrial life is being carried out with radio antennas like the one pictured here. A radio antenna is an example of a system with position controls.

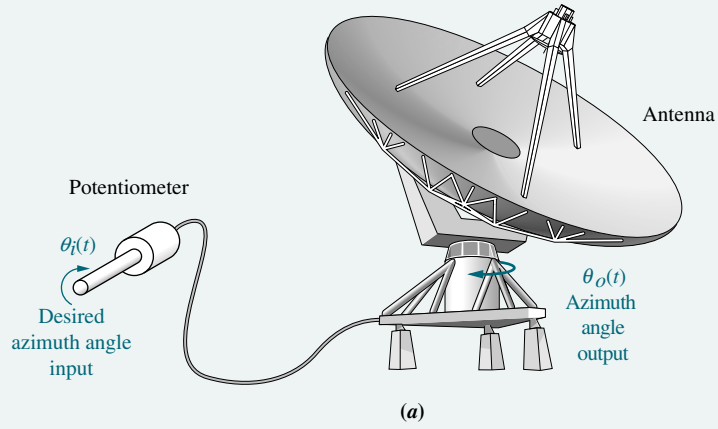
Now that our objectives are stated, how do we meet them? In this section we will look at an example of a feedback control system. The system introduced here will be used in subsequent chapters as a running case study to demonstrate the objectives of those chapters. A colored background like this will identify the case study section at the end of each chapter. Section 1.5, which follows this first case study, explores the design process that will help us build our system.

Antenna Azimuth: An Introduction to Position Control Systems

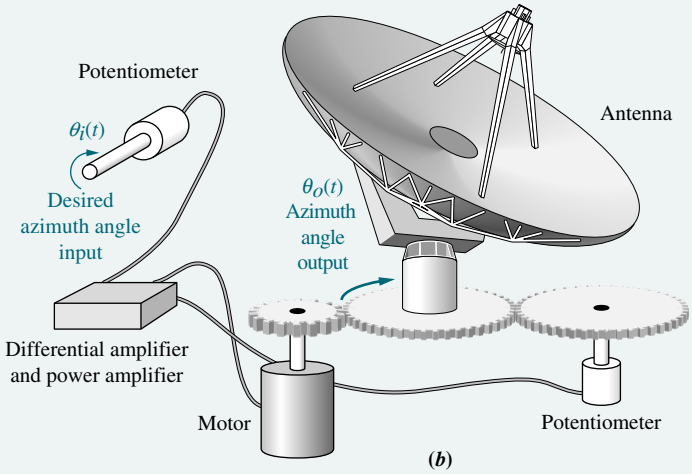
A position control system converts a position input command to a position output response. Position control systems find widespread applications in antennas, robot arms, and computer disk drives. The radio telescope antenna in Figure 1.8 is one example of a system that uses position control systems. In this section, we will look in detail at an antenna azimuth position control system that could be used to position a radio telescope antenna. We will see how the system works and how we can effect changes in its performance. The discussion here will be on a qualitative level, with the objective of getting an intuitive feeling for the systems with which we will be dealing.

An antenna azimuth position control system is shown in Figure 1.9(a), with a more detailed layout and schematic in Figures 1.9(b) and 1.9(c), respectively. Figure 1.9(d) shows *a functional block diagram* of the system. The functions are shown above the blocks, and the required hardware is indicated inside the blocks. Parts of Figure 1.9 are repeated on the front endpapers for future reference.

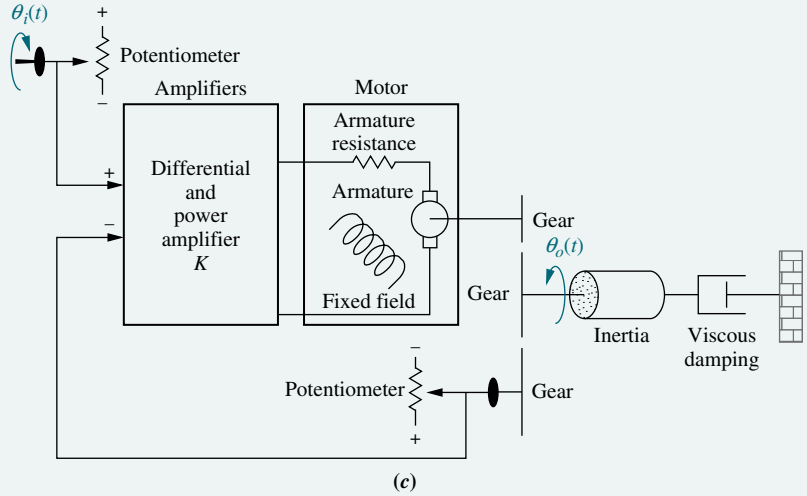
The purpose of this system is to have the azimuth angle output of the antenna, $\theta_o(t)$, follow the input angle of the potentiometer, $\theta_i(t)$. Let us look at Figure 1.9(d) and describe how this system works. The input command is an angular displacement. The potentiometer converts the angular displacement into a voltage.



(*a*)

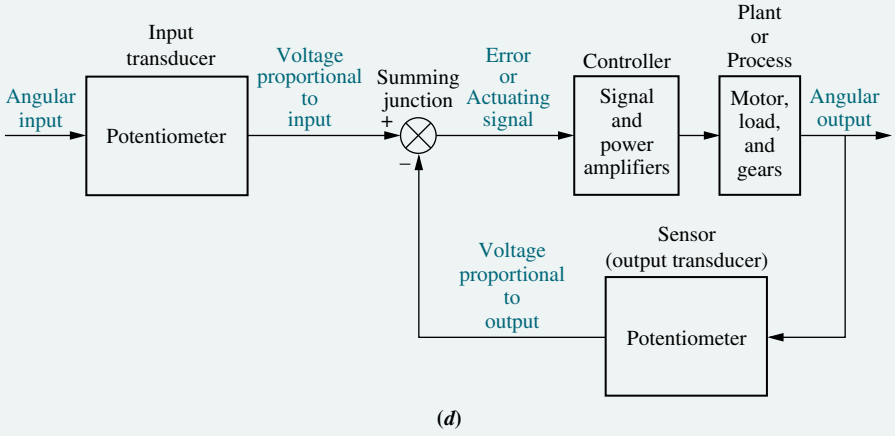


(*b*)



(*c*)

FIGURE 1.9 Antenna azimuth position control system: **a.** system concept; **b.** detailed layout; **c.** schematic; (*figure continues*)



(*d*)

FIGURE 1.9 (*Continued*)
d. functional block diagram

Similarly, the output angular displacement is converted to a voltage by the potentiometer in the feedback path. The signal and power amplifiers boost the difference between the input and output voltages. This amplified actuating signal drives the plant.

The system normally operates to drive the error to zero. When the input and output match, the error will be zero, and the motor will not turn. Thus, the motor is driven only when the output and the input do not match. The greater the difference between the input and the output, the larger the motor input voltage, and the faster the motor will turn.

If we increase the gain of the signal amplifier, will there be an increase in the steady-state value of the output? If the gain is increased, then for a given actuating signal, the motor will be driven harder. However, the motor will still stop when the actuating signal reaches zero, that is, when the output matches the input. The difference in the response, however, will be in the transients. Since the motor is driven harder, it turns faster toward its final position. Also, because of the increased speed, increased momentum could cause the motor to overshoot the final value and be forced by the system to return to the commanded position. Thus, the possibility exists for a transient response that consists of *damped oscillations* (that is, a sinusoidal response whose amplitude diminishes with time) about the steady-state value if the gain is high. The responses for low gain and high gain are shown in Figure 1.10.

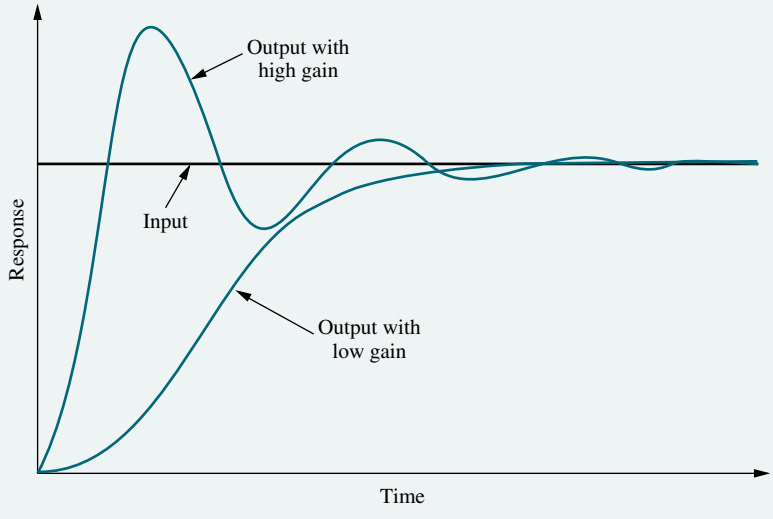


FIGURE 1.10 Response of a position control system, showing effect of high and low controller gain on the output response

We have discussed the transient response of the position control system. Let us now direct our attention to the steady-state position to see how closely the output matches the input after the transients disappear.

We define steady-state error as the difference between the input and the output after the transients have effectively disappeared. The definition holds equally well for step, ramp, and other types of inputs. Typically, the steady-state error decreases with an increase in gain and increases with a decrease in gain. Figure 1.10 shows zero error in the steady-state response; that is, after the transients have disappeared, the output position equals the commanded input position. In some systems, the steady-state error will not be zero; for these systems, a simple gain adjustment to regulate the transient response is either not effective or leads to a trade-off between the desired transient response and the desired steady-state accuracy.

To solve this problem, a controller with a dynamic response, such as an electrical filter, is used along with an amplifier. With this type of controller, it is possible to design both the required transient response and the required steady-state accuracy without the trade-off required by a simple setting of gain. However, the controller is now more complex. The filter in this case is called a compensator. Many systems also use dynamic elements in the feedback path along with the output transducer to improve system performance.

In summary, then, our design objectives and the system's performance revolve around the transient response, the steady-state error, and stability. Gain adjustments can affect performance and sometimes lead to trade-offs between the performance criteria. Compensators can often be designed to achieve performance specifications without the need for trade-offs. Now that we have stated our objectives and some of the methods available to meet those objectives, we describe the orderly progression that leads us to the final system design.

1.5 The Design Process

In this section, we establish an orderly sequence for the design of feedback control systems that will be followed as we progress through the rest of the book. Figure 1.11 shows the described process as well as the chapters in which the steps are discussed.

The antenna azimuth position control system discussed in the last section is representative of control systems that must be analyzed and designed. Inherent in

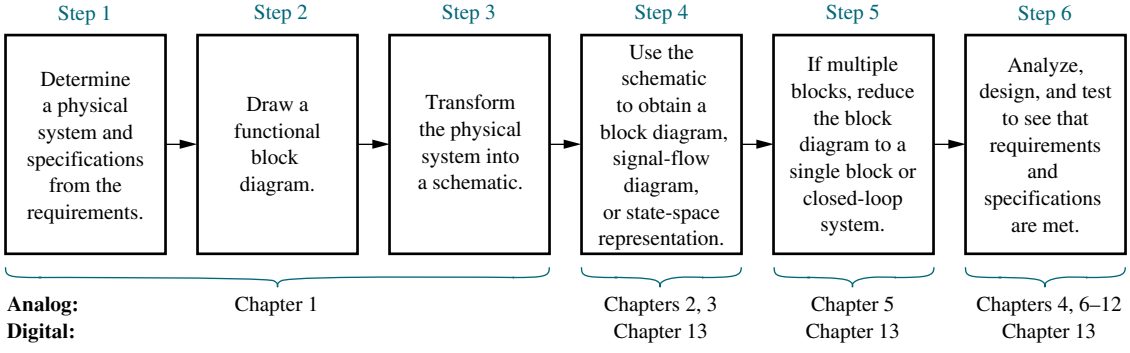


FIGURE 1.11 The control system design process

Figure 1.11 is feedback and communication during each phase. For example, if testing (Step 6) shows that requirements have not been met, the system must be redesigned and retested. Sometimes requirements are conflicting and the design cannot be attained. In these cases, the requirements have to be respecified and the design process repeated. Let us now elaborate on each block of Figure 1.11.

Step 1: Transform Requirements Into a Physical System

We begin by transforming the requirements into a physical system. For example, in the antenna azimuth position control system, the requirements would state the desire to position the antenna from a remote location and describe such features as weight and physical dimensions. Using the requirements, design specifications, such as desired transient response and steady-state accuracy, are determined. Perhaps an overall concept, such as Figure 1.9(*a*), would result.

Step 2: Draw a Functional Block Diagram

The designer now translates a qualitative description of the system into a functional block diagram that describes the component parts of the system (that is, function and/or hardware) and shows their interconnection. Figure 1.9(*d*) is an example of a functional block diagram for the antenna azimuth position control system. It indicates functions such as input transducer and controller, as well as possible hardware descriptions such as amplifiers and motors. At this point the designer may produce a detailed layout of the system, such as that shown in Figure 1.9(*b*), from which the next phase of the analysis and design sequence, developing a schematic diagram, can be launched.

Step 3: Create a Schematic

As we have seen, position control systems consist of electrical, mechanical, and electromechanical components. After producing the description of a physical system, the control systems engineer transforms the physical system into a schematic diagram. The control system designer can begin with the physical description, as contained in Figure 1.9(*a*), to derive a schematic. The engineer must make approximations about the system and neglect certain phenomena, or else the schematic will be unwieldy, making it difficult to extract a useful mathematical model during the next phase of the analysis and design sequence. The designer starts with a simple schematic representation and, at subsequent phases of the analysis and design sequence, checks the assumptions made about the physical system through analysis and computer simulation. If the schematic is too simple and does not adequately account for observed behavior, the control systems engineer adds phenomena to the schematic that were previously assumed negligible. A schematic diagram for the antenna azimuth position control system is shown in Figure 1.9(*c*).

When we draw the potentiometers, we make our first simplifying assumption by neglecting their friction or inertia. These mechanical characteristics yield a dynamic, rather than an instantaneous, response in the output voltage. We assume that these mechanical effects are negligible and that the voltage across a potentiometer changes instantaneously as the potentiometer shaft turns.

A differential amplifier and a power amplifier are used as the controller to yield gain and power amplification, respectively, to drive the motor. Again, we assume that the dynamics of the amplifiers are rapid compared to the response time of the motor; thus, we model them as a pure gain, K.

A dc motor and equivalent load produce the output angular displacement. The speed of the motor is proportional to the voltage applied to the motor's *armature circuit*. Both inductance and resistance are part of the armature circuit. In showing

just the armature resistance in Figure 1.9(c), we assume the effect of the armature inductance is negligible for a dc motor.

The designer makes further assumptions about the load. The load consists of a rotating mass and bearing friction. Thus, the model consists of *inertia* and *viscous damping* whose resistive torque increases with speed, as in an automobile's shock absorber or a screen door damper.

The decisions made in developing the schematic stem from knowledge of the physical system, the physical laws governing the system's behavior, and *practical experience*. These decisions are not easy; however, as you acquire more design experience, you will gain the insight required for this difficult task.

Step 4: Develop a Mathematical Model (Block Diagram)

Once the schematic is drawn, the designer uses physical laws, such as Kirchhoff's laws for electrical networks and Newton's law for mechanical systems, along with simplifying assumptions, to model the system mathematically. These laws are

Kirchhoff's voltage law	The sum of voltages around a closed path equals zero.
Kirchhoff's current law	The sum of electric currents flowing from a node equals zero.
Newton's laws	The sum of forces on a body equals zero;[3] the sum of moments on a body equals zero.

Kirchhoff's and Newton's laws lead to mathematical models that describe the relationship between the input and output of dynamic systems. One such model is the *linear, time-invariant differential equation*, Eq. (1.2):

$$\frac{d^m c(t)}{dt^n} + d_{n-1}\frac{d^{m-1}c(t)}{dt^{n-1}} + \cdots + d_0 c(t) = b_m \frac{d^m r(t)}{dt^m} + b_{m-1}\frac{d^{m-1}r(t)}{dt^{m-1}} + \cdots + b_0 r(t) \tag{1.2}$$

[4]

Many systems can be approximately described by this equation, which relates the output, $c(t)$, to the input, $r(t)$, by way of the system parameters, a_i and b_j. We assume the reader is familiar with differential equations. Problems and a bibliography are provided at the end of the chapter for you to review this subject.

Simplifying assumptions made in the process of obtaining a mathematical model usually leads to a low-order form of Eq. (1.2). Without the assumptions the system model could be of high order or described with nonlinear, time-varying, or partial differential equations. These equations complicate the design process and reduce the designer's insight. Of course, all assumptions must be checked and all simplifications justified through analysis or testing. If the assumptions for simplification cannot be justified, then the model cannot be simplified. We examine some of these simplifying assumptions in Chapter 2.

In addition to the differential equation, the *transfer function* is another way of mathematically modeling a system. The model is derived from the linear, time-invariant differential equation using what we call the *Laplace transform*. Although the transfer

[3] Alternately, $\sum$ forces $= Ma$. In this text the force, Ma, will be brought to the left-hand side of the equation to yield $\sum$ forces $= 0$ (D'Alembert's principle). We can then have a consistent analogy between force and voltage, and Kirchhoff's and Newton's laws (that is, $\sum$ forces $= 0$; $\sum$ voltages $= 0$).

[4] The right-hand side of Eq. (1.2) indicates differentiation of the input, $r(t)$. In physical systems, differentiation of the input introduces noise. In Chapters 3 and 5 we show implementations and interpretations of Eq. (1.2) that do not require differentiation of the input.

function can be used only for linear systems, it yields more intuitive information than the differential equation. We will be able to change system parameters and rapidly sense the effect of these changes on the system response. The transfer function is also useful in modeling the interconnection of subsystems by forming a block diagram similar to Figure 1.9(*d*) but with a mathematical function inside each block.

Still another model is the *state-space representation*. One advantage of state-space methods is that they can also be used for systems that cannot be described by linear differential equations. Further, state-space methods are used to model systems for simulation on the digital computer. Basically, this representation turns an *n*th-order differential equation into *n* simultaneous first-order differential equations. Let this description suffice for now; we describe this approach in more detail in Chapter 3.

Finally, we should mention that to produce the mathematical model for a system, we require knowledge of the parameter values, such as equivalent resistance, inductance, mass, and damping, which is often not easy to obtain. Analysis, measurements, or specifications from vendors are sources that the control systems engineer may use to obtain the parameters.

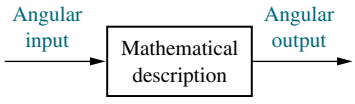


FIGURE 1.12 Equivalent block diagram for the antenna azimuth position control system

Step 5: Reduce the Block Diagram

Subsystem models are interconnected to form block diagrams of larger systems, as in Figure 1.9(*d*), where each block has a mathematical description. Notice that many signals, such as proportional voltages and error, are internal to the system. There are also two signals—angular input and angular output—that are external to the system. In order to evaluate system response in this example, we need to reduce this large system's block diagram to a single block with a mathematical description that represents the system from its input to its output, as shown in Figure 1.12. Once the block diagram is reduced, we are ready to analyze and design the system.

Step 6: Analyze and Design

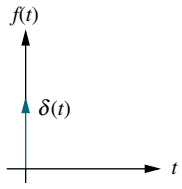
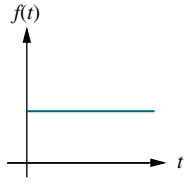
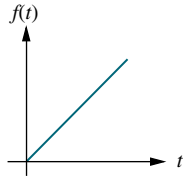
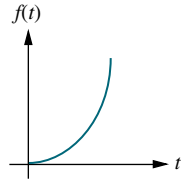
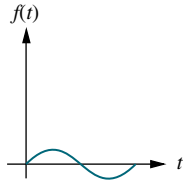
The next phase of the process, following block diagram reduction, is analysis and design. If you are interested only in the performance of an individual subsystem, you can skip the block diagram reduction and move immediately into analysis and design. In this phase, the engineer analyzes the system to see if the response specifications and performance requirements can be met by simple adjustments of system parameters. If specifications cannot be met, the designer then designs additional hardware in order to effect a desired performance.

Test input signals are used, both analytically and during testing, to verify the design. It is neither necessarily practical nor illuminating to choose complicated input signals to analyze a system's performance. Thus, the engineer usually selects standard test inputs. These inputs are impulses, steps, ramps, parabolas, and sinusoids, as shown in Table 1.1.

An *impulse* is infinite at $t = 0$ and zero elsewhere. The area under the unit impulse is 1. An approximation of this type of waveform is used to place initial energy into a system so that the response due to that initial energy is only the transient response of a system. From this response the designer can derive a mathematical model of the system.

A *step* input represents a *constant command*, such as position, velocity, or acceleration. Typically, the step input command is of the same form as the output. For example, if the system's output is position, as it is for the antenna azimuth position control system, the step input represents a desired position, and the output represents the actual position. If the system's output is velocity, as is the spindle speed for a video disc player, the step input represents a constant desired speed, and the output represents the actual speed. The designer uses step inputs because both the transient response and the steady-state response are clearly visible and can be evaluated.

TABLE 1.1 Test waveforms used in control systems

Input	Function	Description	Sketch	Use
Impulse	$\delta(t)$	$\delta(t) = \infty$ for $0- < t < 0+$ $= 0$ elsewhere $\int_{0-}^{0+} \delta(t)dt = 1$		Transient response Modeling
Step	$u(t)$	$u(t) = 1$ for $t > 0$ $= 0$ for $t < 0$		Transient response Steady-state error
Ramp	$tu(t)$	$tu(t) = t$ for $t \geq 0$ $= 0$ elsewhere		Steady-state error
Parabola	$\frac{1}{2}t^2u(t)$	$\frac{1}{2}t^2u(t) = \frac{1}{2}t^2$ for $t \geq 0$ $= 0$ elsewhere		Steady-state error
Sinusoid	$\sin \omega t$			Transient response Modeling Steady-state error

The *ramp* input represents a *linearly increasing command*. For example, if the system's output is position, the input ramp represents a linearly increasing position, such as that found when tracking a satellite moving across the sky at constant speed. If the system's output is velocity, the input ramp represents a linearly increasing velocity. The response to an input ramp test signal yields additional information about the steady-state error. The previous discussion can be extended to *parabolic* inputs, which are also used to evaluate a system's steady-state error.

Sinusoidal inputs can also be used to test a physical system to arrive at a mathematical model. We discuss the use of this waveform in detail in Chapters 10 and 11.

We conclude that one of the basic analysis and design requirements is to evaluate the time response of a system for a given input. Throughout the book you will learn numerous methods for accomplishing this goal.

The control systems engineer must take into consideration other characteristics about feedback control systems. For example, control system behavior is altered by fluctuations in component values or system parameters. These variations can be

caused by temperature, pressure, or other environmental changes. Systems must be built so that expected fluctuations do not degrade performance beyond specified bounds. A *sensitivity* analysis can yield the percentage of change in a specification as a function of a change in a system parameter. One of the designer's goals, then, is to build a system with minimum sensitivity over an expected range of environmental changes.

In this section we looked at some control systems analysis and design considerations. We saw that the designer is concerned about transient response, steady-state error, stability, and sensitivity. The text pointed out that although the basis of evaluating system performance is the differential equation, other methods, such as transfer functions and state space, will be used. The advantages of these new techniques over differential equations will become apparent as we discuss them in later chapters.

1.6 Computer-Aided Design

Now that we have discussed the analysis and design sequence, let us discuss the use of the computer as a computational tool in this sequence. The computer plays an important role in the design of modern control systems. In the past, control system design was labor intensive. Many of the tools we use today were implemented through hand calculations or, at best, using plastic graphical aid tools. The process was slow, and the results not always accurate. Large mainframe computers were then used to simulate the designs.

Today we are fortunate to have computers and software that remove the drudgery from the task. At our own desktop computers, we can perform analysis, design, and simulation with one program. With the ability to simulate a design rapidly, we can easily make changes and immediately test a new design. We can play what-if games and try alternate solutions to see if they yield better results, such as reduced sensitivity to parameter changes. We can include nonlinearities and other effects and test our models for accuracy.

MATLAB

The computer is an integral part of modern control system design, and many computational tools are available for your use. In this book we use MATLAB and the MATLAB Control System Toolbox, which expands MATLAB to include control system–specific commands. In addition, presented are several MATLAB enhancements that give added functionality to MATLAB and the Control Systems Toolbox. Included are (1) Simulink, which uses a graphical user interface (GUI); (2) the LTI Viewer, which permits measurements to be made directly from time and frequency response curves; (3) the SISO Design Tool, a convenient and intuitive analysis and design tool; and (4) the Symbolic Math Toolbox, which saves labor when making symbolic calculations required in control system analysis and design. Some of these enhancements may require additional software available from The MathWorks, Inc.

MATLAB is presented as an alternate method of solving control system problems. You are encouraged to solve problems first by hand and then by MATLAB so that insight is not lost through mechanized use of computer programs. To this end, many examples throughout the book are solved by hand, followed by suggested use of MATLAB.

As an enticement to begin using MATLAB, simple program statements that you can try are suggested throughout the chapters at appropriate locations. Throughout the book, various icons appear in the margin to identify MATLAB references that direct you to the proper program in the proper appendix and tell you what you

will learn. Selected end-of-chapter problems and Case Study Challenges to be solved using MATLAB have also been marked with appropriate icons. The following list itemizes the specific components of MATLAB used in this book, the icon used to identify each, and the appendix in which a description can be found:

MATLAB



MATLAB/Control System Toolbox tutorials and code are found in Appendix B and identified in the text with the MATLAB icon shown in the margin.

Simulink



Simulink tutorials and diagrams are found in Appendix C and identified in the text with the Simulink icon shown in the margin.

Gui Tool



MATLAB GUI tools, tutorials, and examples are in Appendix E at www.wiley.com/college/nise and identified in the text with the GUI Tool icon shown in the margin. These tools consist of the LTI Viewer and the SISO Design Tool.

Symbolic Math



Symbolic Math Toolbox tutorials and code are found in Appendix F at www.wiley.com/college/nise and identified in the text with the Symbolic Math icon shown in the margin.

MATLAB code itself is not platform specific. The same code runs on PCs and workstations that support MATLAB. Although there are differences in installing and managing MATLAB files, we do not address them in this book. Also, there are many more commands in MATLAB and the MATLAB toolboxes than are covered in the appendixes. Please explore the bibliographies at the end of the applicable appendixes to find out more about MATLAB file management and MATLAB instructions not covered in this textbook.

LabVIEW

LabVIEW is a programming environment presented as an alternative to MATLAB. This graphical alternative produces front panels of virtual instruments on your computer that are pictorial reproductions of hardware instruments, such as waveform generators or oscilloscopes. Underlying the front panels are block diagrams. The blocks contain underlying code for the controls and indicators on the front panel. Thus, a knowledge of coding is not required. Also, parameters can be easily passed or viewed from the front panel.

LabVIEW



A LabVIEW tutorial is in Appendix D and all LabVIEW material is identified with the LabVIEW icon shown in the margin.

You are encouraged to use computational aids throughout this book. Those not using MATLAB or LabVIEW should consult Appendix H at www.wiley.com/college/nise for a discussion of other alternatives. Now that we have introduced control systems to you and established a need for computational aids to perform analysis and design, we will conclude with a discussion of your career as a control systems engineer and look at the opportunities and challenges that await you.

1.7 The Control Systems Engineer

Control systems engineering is an exciting field in which to apply your engineering talents, because it cuts across numerous disciplines and numerous functions within those disciplines. The control engineer can be found at the top level of large projects, engaged at the conceptual phase in determining or implementing

overall system requirements. These requirements include total system performance specifications, subsystem functions, and the interconnection of these functions, including interface requirements, hardware and software design, and test plans and procedures.

Many engineers are engaged in only one area, such as circuit design or software development. However, as a control systems engineer, you may find yourself working in a broad arena and interacting with people from numerous branches of engineering and the sciences. For example, if you are working on a biological system, you will need to interact with colleagues in the biological sciences, mechanical engineering, electrical engineering, and computer engineering, not to mention mathematics and physics. You will be working with these engineers at all levels of project development from concept through design and, finally, testing. At the design level, the control systems engineer can be performing hardware selection, design, and interface, including total subsystem design to meet specified requirements. The control engineer can be working with sensors and motors as well as electronic, pneumatic, and hydraulic circuits.

The space shuttle provides another example of the diversity required of the systems engineer. In the previous section, we showed that the space shuttle's control systems cut across many branches of science: orbital mechanics and propulsion, aerodynamics, electrical engineering, and mechanical engineering. Whether or not you work in the space program, as a control systems engineer you will apply broad-based knowledge to the solution of engineering control problems. You will have the opportunity to expand your engineering horizons beyond your university curriculum.

You are now aware of future opportunities. But for now, what advantages does this course offer to a student of control systems (other than the fact that you need it to graduate)? Engineering curricula tend to emphasize *bottom-up* design. That is, you start from the components, develop circuits, and then assemble a product. In *top-down* design, a high-level picture of the requirements is first formulated; then the functions and hardware required to implement the system are determined. You will be able to take a top-down systems approach as a result of this course.

A major reason for not teaching top-down design throughout the curriculum is the high level of mathematics initially required for the systems approach. For example, control systems theory, which requires differential equations, could not be taught as a lower-division course. However, while progressing through bottom-up design courses, it is difficult to see how such design fits logically into the large picture of the product development cycle.

After completing this control systems course, you will be able to stand back and see how your previous studies fit into the large picture. Your amplifier course or vibrations course will take on new meaning as you begin to see the role design work plays as part of product development. For example, as engineers, we want to describe the physical world mathematically so that we can create systems that will benefit humanity. You will find that you have indeed acquired, through your previous courses, the ability to model physical systems mathematically, although at the time you might not have understood where in the product development cycle the modeling fits. This course will clarify the analysis and design procedures and show you how the knowledge you acquired fits into the total picture of system design.

Understanding control systems enables students from all branches of engineering to speak a common language and develop an appreciation and working knowledge of the other branches. You will find that there really is not much difference among the branches of engineering as far as the goals and applications are concerned. As you study control systems, you will see this commonality.

Summary

Control systems contribute to every aspect of modern society. In our homes we find them in everything from toasters to heating systems to VCRs. Control systems also have widespread applications in science and industry, from steering ships and planes to guiding missiles and the space shuttle. Control systems also exist naturally; our bodies contain numerous control systems. Even economic and psychological system representations have been proposed based on control system theory. Control systems are used where power gain, remote control, or conversion of the form of the input is required.

A control system has an *input*, a *process*, and an *output*. Control systems can be *open loop* or *closed loop*. Open-loop systems do not monitor or correct the output for disturbances; however, they are simpler and less expensive than closed-loop systems. Closed-loop systems monitor the output and compare it to the input. If an error is detected, the system corrects the output and hence corrects the effects of disturbances.

Control systems analysis and design focuses on three primary objectives:

1. Producing the desired transient response
2. Reducing steady-state errors
3. Achieving stability

A system must be stable in order to produce the proper transient and steady-state response. Transient response is important because it affects the speed of the system and influences human patience and comfort, not to mention mechanical stress. Steady-state response determines the accuracy of the control system; it governs how closely the output matches the desired response.

The design of a control system follows these steps:

Step 1 Determine a physical system and specifications from requirements.

Step 2 Draw a functional block diagram.

Step 3 Represent the physical system as a schematic.

Step 4 Use the schematic to obtain a mathematical model, such as a block diagram.

Step 5 Reduce the block diagram.

Step 6 Analyze and design the system to meet specified requirements and specifications that include stability, transient response, and steady-state performance.

In the next chapter we continue through the analysis and design sequence and learn how to use the schematic to obtain a mathematical model.

Review Questions

1. Name three applications for feedback control systems.
2. Name three reasons for using feedback control systems and at least one reason for not using them.
3. Give three examples of open-loop systems.
4. Functionally, how do closed-loop systems differ from open-loop systems?
5. State one condition under which the error signal of a feedback control system would not be the difference between the input and the output.

6. If the error signal is not the difference between input and output, by what general name can we describe the error signal?
7. Name two advantages of having a computer in the loop.
8. Name the three major design criteria for control systems.
9. Name the two parts of a system's response.
10. Physically, what happens to a system that is unstable?
11. Instability is attributable to what part of the total response?
12. Describe a typical control system analysis task.
13. Describe a typical control system design task.
14. Adjustments of the forward path gain can cause changes in the transient response. True or false?
15. Name three approaches to the mathematical modeling of control systems.
16. Briefly describe each of your answers to Question 15.

Problems

1. A variable resistor, called a *potentiometer*, is shown in Figure P1.1. The resistance is varied by moving a wiper arm along a fixed resistance. The resistance from A to C is fixed, but the resistance from B to C varies with the position of the wiper arm. If it takes 10 turns to move the wiper arm from A to C, draw a block diagram of the potentiometer showing the input variable, the output variable, and (inside the block) the gain, which is a constant and is the amount by which the input is multiplied to obtain the output. [Section 1.4: Introduction to a Case Study]

WileyPLUS [5] WPCS Control Solutions

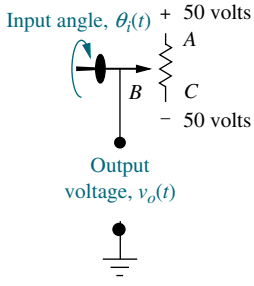


FIGURE P1.1 Potentiometer

2. A temperature control system operates by sensing the difference between the thermostat setting and the actual temperature and then opening a fuel valve an amount proportional to this difference. Draw a functional closed-loop block diagram similar to Figure 1.9(d) identifying the input and output transducers, the controller, and the plant. Further, identify the input and output signals of all subsystems previously described. [Section 1.4: Introduction to a Case Study]
3. An aircraft's attitude varies in roll, pitch, and yaw as defined in Figure P1.2. Draw a functional block diagram for a closed-loop system that stabilizes the roll as follows: The system measures the actual roll angle with a gyro and compares the actual roll angle with the desired roll angle. The ailerons respond to the roll-angle error by undergoing an angular deflection. The aircraft responds to this angular deflection, producing a roll angle rate. Identify the input and output transducers, the controller, and the plant. Further, identify the nature of each signal. [Section 1.4: Introduction to a Case Study]

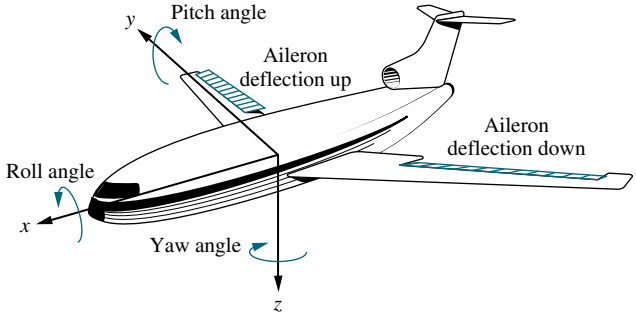


FIGURE P1.2 Aircraft attitude defined

[5] The WileyPLUS icon identifies interactive worked examples and problems. These problems, developed by JustAsk, are worked in detail and offer explanations of every facet of the solution. The identified examples and problems can be accessed at www.wiley.com/college/nise.

4. Many processes operate on rolled material that moves from a supply reel to a take-up reel. Typically, these systems, called *winders*, control the material so that it travels at a constant velocity. Besides velocity, complex winders also control tension, compensate for roll inertia while accelerating or decelerating, and regulate acceleration due to sudden changes. A winder is shown in Figure P1.3. The force transducer measures tension; the winder pulls against the nip rolls, which provide an opposing force; and the bridle provides slip. In order to compensate for changes in speed, the material is looped around a *dancer*. The loop prevents rapid changes from causing excessive slack or damaging the material. If the dancer position is sensed by a potentiometer or other device, speed variations due to buildup on the take-up reel or other causes can be controlled by comparing the potentiometer voltage to the commanded speed. The system then corrects the speed and resets the dancer to the desired position (*Ayers*, *1988*). Draw a functional block diagram for the speed control system, showing each component and signal. [Section 1.4: Introduction to a Case Study]

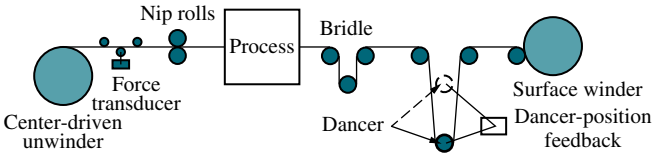


FIGURE P1.3 Winder

5. In a nuclear power generating plant, heat from a reactor is used to generate steam for turbines. The rate of the fission reaction determines the amount of heat generated, and this rate is controlled by rods inserted into the radioactive core. The rods regulate the flow of neutrons. If the rods are lowered into the core, the rate of fission will diminish; if the rods are raised, the fission rate will increase. By automatically controlling the position of the rods, the amount of heat generated by the reactor can be regulated. Draw a functional block diagram for the nuclear reactor control system shown in Figure P1.4. Show all blocks and signals. [Section 1.4: Introduction to a Case Study]

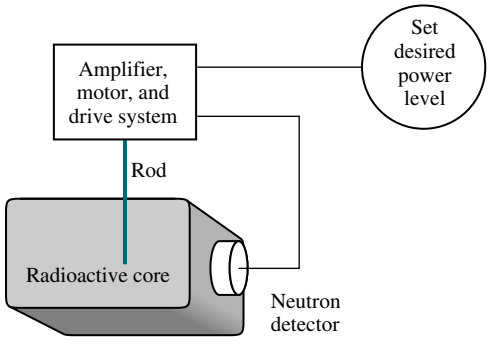


FIGURE P1.4 Control of a nuclear reactor

6. A university wants to establish a control system model that represents the student population as an output, with the desired student population as an input. The administration determines the rate of admissions by comparing the current and desired student populations. The admissions office then uses this rate to admit students. Draw a functional block diagram showing the administration and the admissions office as blocks of the system. Also show the following signals: the desired student population, the actual student population, the desired student rate as determined by the administration, the actual student rate as generated by the admissions office, the dropout rate, and the net rate of influx. [Section 1.4: Introduction to a Case Study]

7. We can build a control system that will automatically adjust a motorcycle's radio volume as the noise generated by the motorcycle changes. The noise generated by the motorcycle increases with speed. As the noise increases, the system increases the volume of the radio. Assume that the amount of noise can be represented by a voltage generated by the speedometer cable, and the volume of the radio is controlled by a dc voltage (*Hogan*, *1988*). If the dc voltage represents the desired volume disturbed by the motorcycle noise, draw the functional block diagram of the automatic volume control system, showing the input transducer, the volume control circuit, and the speed transducer as blocks. Also show the following signals: the desired volume as an input, the actual volume as an output, and voltages representing speed, desired volume, and actual volume. [Section 1.4: Introduction to a Case Study]

WileyPLUS
WPCS
Control Solutions

8. Your bathtub at home is a control system that keeps the water level constant. A constant flow from the tap yields a constant water level, because the flow rate through the drain increases as the water level increases, and decreases as the water level decreases. After equilibrium has been reached, the level can be controlled by controlling the input flow rate. A low input flow rate yields a lower level, while a higher input flow rate yields a higher level. [Section 1.4: Introduction to a Case Study]

a. Sketch a control system that uses this principle to precisely control the fluid level in a tank. Show the intake and drain valves, the tank, any sensors and transducers, and the interconnection of all components.

b. Draw a functional block diagram of the system, identifying the input and output signals of each block.

9. A dynamometer is a device used to measure torque and speed and to vary the load on rotating devices. The dynamometer operates as follows to control the amount of torque: A hydraulic actuator attached to the axle presses a tire against a rotating flywheel. The greater the displacement of the actuator, the more force that is applied to the rotating flywheel. A strain gage load cell senses the force. The displacement of the actuator is controlled by an electrically operated valve whose displacement regulates fluid flowing into the actuator (*D'Souza*, *1988*). Draw a functional block diagram of a closed-loop system that uses the described dynamometer to regulate the force against the tire during testing. Show all signals and systems. Include amplifiers that power the valve, the valve, the actuator and load, and the tire. [Section 1.4: Introduction to a Case Study]

10. During a medical operation an anesthesiologist controls the depth of unconsciousness by controlling the concentration of isoflurane in a vaporized mixture with oxygen and nitrous oxide. The depth of anesthesia is measured by the patient's blood pressure. The anesthesiologist also regulates ventilation, fluid balance, and the administration of other drugs. In order to free the anesthesiologist to devote more time to the latter tasks, and in the interest of the patient's safety, we wish to automate the depth of anesthesia by automating the control of isoflurane concentration. Draw a functional block diagram of the system showing pertinent signals and subsystems (*Meier*, *1992*). [Section 1.4: Introduction to a Case Study]

WileyPLUS WPCS Control Solutions

11. The vertical position, $x(t)$, of the grinding wheel shown in Figure P1.5 is controlled by a closed-loop system. The input to the system is the desired depth of grind, and the output is the actual depth of grind. The difference between the desired depth and the actual depth drives the motor, resulting in a force applied to the work. This force results in a feed velocity for the grinding wheel (*Jenkins*, *1997*). Draw a closed-loop functional block diagram for the grinding process, showing the input, output, force, and grinder feed rate. [Section 1.4: Introduction to a Case Study]

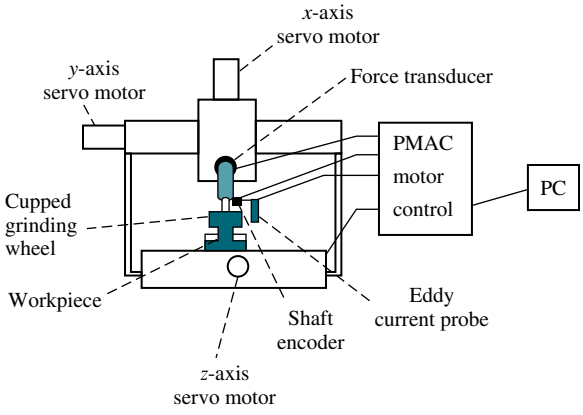


FIGURE P1.5 Grinder system (Reprinted with permission of ASME.)

12. A high-speed proportional solenoid valve is shown in Figure P1.6. A voltage proportional to the desired position of the spool is applied to the coil. The resulting magnetic field produced by the current in the coil causes the armature to move. A push pin connected to the armature moves the spool. A linear voltage differential transformer (LVDT) that outputs a voltage proportional to displacement senses the spool's position. This voltage can be used in a feedback path to implement closed-loop operation (*Vaughan*, *1996*). Draw a functional block diagram of the valve, showing input and output positions, coil voltage, coil current, and spool force. [Section 1.4: Introduction to a Case Study]

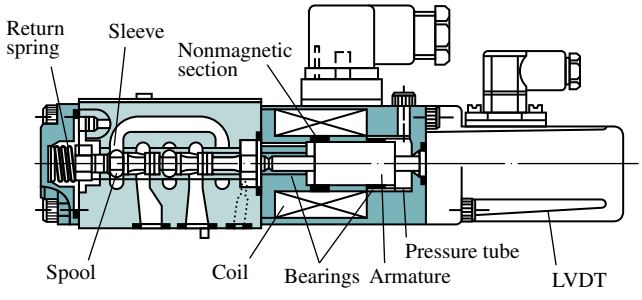


FIGURE P1.6 High-speed proportional solenoid valve (Reprinted with permission of ASME.)

13. The human eye has a biological control system that varies the pupil diameter to maintain constant light

intensity to the retina. As the light intensity increases, the optical nerve sends a signal to the brain, which commands internal eye muscles to decrease the pupil's eye diameter. When the light intensity decreases, the pupil diameter increases.

a. Draw a functional block diagram of the light-pupil system indicating the input, output, and intermediate signals; the sensor; the controller; and the actuator. [Section 1.4: Introduction to a Case Study]

b. Under normal conditions the incident light will be larger than the pupil, as shown in Figure P1.7(*a*). If the incident light is smaller than the diameter of the pupil as shown in Figure P1.7(*b*), the feedback path is broken (*Bechhoefer*, *2005*). Modify your block diagram from Part **a.** to show where the loop is broken. What will happen if the narrow beam of light varies in intensity, say in a sinusoidal fashion?

c. It has been found (*Bechhoefer*, *2005*) that it takes the pupil about 300 milliseconds to react to a change in the incident light. If light shines off center to the retina as shown in Figure P1.7(*c*), describe the response of the pupil with delay present and then without delay present.

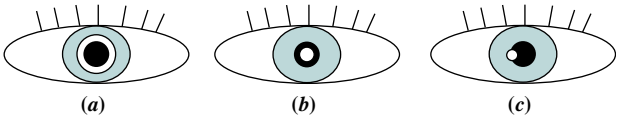


FIGURE P1.7 Pupil is shown black; light beam is shown white. **a.** Light beam diameter is larger than pupil. **b.** Light beam diameter is smaller than pupil. **c.** Narrow light beam is illuminated at pupil's edge.

14. A Segway®[6] Personal Transporter (PT) (Figure P1.8) is a two-wheeled vehicle in which the human operator stands vertically on a platform. As the driver leans left, right, forward, or backward, a set of sensitive gyroscopic sensors sense the desired input. These signals are fed to a computer that amplifies them and commands motors to propel the vehicle in the desired direction. One very important feature of the PT is its safety: The system will maintain its vertical position within a specified angle despite road disturbances, such as uphills and downhills or even if the operator over-leans in any direction. Draw a functional block diagram of the PT system that keeps the system in a vertical position. Indicate the input and output signals, intermediate signals, and main subsystems. (http://segway.com)



FIGURE P1.8 The Segway Personal Transporter (PT)

15. In humans, hormone levels, alertness, and core body temperature are synchronized through a 24-hour circadian cycle. Daytime alertness is at its best when sleep/wake cycles are in synch with the circadian cycle. Thus alertness can be easily affected with a distributed work schedule, such as the one to which astronauts are subjected. It has been shown that the human circadian cycle can be delayed or advanced through light stimulus. To ensure optimal alertness, a system is designed to track astronauts' circadian cycles and increase the quality of sleep during missions. Core body temperature can be used as an indicator of the circadian cycle. A computer model with optimum circadian body temperature variations can be compared to an astronaut's body temperatures. Whenever a difference is detected, the astronaut is subjected to a light stimulus to advance or delay the astronaut's circadian cycle (*Mott*, *2003*). Draw a functional block diagram of the system. Indicate the input and output signals, intermediate signals, and main subsystems.

WileyPLUS WPCS Control Solutions

16. Tactile feedback is an important component in the learning of motor skills such as dancing, sports, and physical rehabilitation. A suit with white dots recognized by a vision system to determine arm joint positions with millimetric precision was developed. This suit is worn by both teacher and

[6] Segway is a registered trademark of Segway, Inc. in the United States and/or other countries.

student to provide position information. (*Lieberman, 2007*). If there is a difference between the teacher's positions and that of the student, vibrational feedback is provided to the student through 8 strategically placed vibrotactile actuators in the wrist and arm, which take advantage of a sensory effect known as *cutaneous rabbit* that tricks the subject to feel uniformly spaced stimuli in places where the actuators are not present. These stimuli help the student adjust to correct the motion. In summary, the system consists of an instructor and a student having their movements followed by the vision system. Their movements are fed into a computer that finds the differences between their joint positions and provides proportional vibrational strength feedback to the student. Draw a block diagram describing the system design.

17. Given the electric network shown in Figure P1.9. [Review]

WileyPLUS WPCS Control Solutions

a. Write the differential equation for the network if $v(t) = u(t)$, a unit step.

b. Solve the differential equation for the current, $i(t)$, if there is no initial energy in the network.

c. Make a plot of your solution if $R/L = 1$.

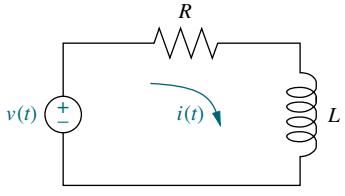


FIGURE P1.9 *RL* network

18. Repeat Problem 17 using the network shown in Figure P1.10. Assume $R = 2\,\Omega$, $L = 1$ H, and $1/LC = 25$. [Review]

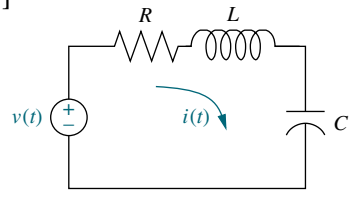


FIGURE P1.10 *RLC* network

19. Solve the following differential equations using classical methods. Assume zero initial conditions. [Review]

WileyPLUS WPCS Control Solutions

a. $\dfrac{dx}{dt} + 7x = 5\cos 2t$

b. $\dfrac{d^2x}{dt^2} + 6\dfrac{dx}{dt} + 8x = 5\sin 3t$

c. $\dfrac{d^2x}{dt^2} + 8\dfrac{dx}{dt} + 25x = 10u(t)$

20. Solve the following differential equations using classical methods and the given initial conditions: [Review]

WileyPLUS WPCS Control Solutions

a. $\dfrac{d^2x}{dt^2} + 2\dfrac{dx}{dt} + 2x = \sin 2t$

$x(0) = 2;\ \dfrac{dx}{dt}(0) = -3$

b. $\dfrac{d^2x}{dt^2} + 2\dfrac{dx}{dt} + x = 5e^{-2t} + t$

$x(0) = 2;\ \dfrac{dx}{dt}(0) = 1$

c. $\dfrac{d^2x}{dt^2} + 4x = t^2$

$x(0) = 1;\ \dfrac{dx}{dt}(0) = 2$

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

21. High-speed rail pantograph. Some high-speed rail systems are powered by electricity supplied to a pantograph on the train's roof from a catenary overhead, as shown in Figure P1.11. The force applied by the pantograph to the catenary is regulated to avoid loss of contact due to excessive transient motion. A proposed method to regulate the force uses a closed-loop feedback system, whereby a force, F_{up}, is applied to the bottom of the pantograph, resulting in an output

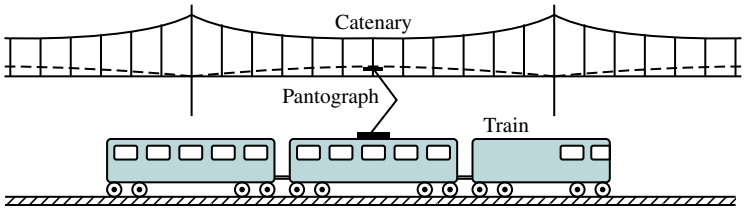


FIGURE P1.11 High-speed rail system showing pantograph and catenary (Reprinted with permission of ASME.)

force applied to the catenary at the top. The contact between the head of the pantograph and the catenary is represented by a spring. The output force is proportional to the displacement of this spring, which is the difference between the catenary and pantograph head vertical positions (*O'Connor, 1997*). Draw a functional block diagram showing the following signals: the desired output force as the input; the force, F_{up}, applied to the bottom of the pantograph; the difference in displacement between the catenary and pantograph head; and the output contact force. Also, show blocks representing the input transducer, controller, actuator generating F_{up}, pantograph dynamics, spring described above, and output sensor. All forces and displacements are measured from equilibrium.

22. Contol of HIV/AIDS. As of 2005, the number of people living worldwide with Human Immunodeficiency Virus/Acquired Immune Deficiency Syndrome (HIV/AIDS) was estimated at 40 million, with 5 million new infections per year and 3 million deaths due to the disease (*UNAIDS, 2005*). Currently there is no known cure for the disease, and HIV cannot be completely eliminated in an infected individual. Drug combinations can be used to maintain the virus numbers at low levels, which helps prevent AIDS from developing. A common treatment for HIV is the administration of two types of drugs: reverse transcriptase inhibitors (RTIs) and protease inhibitors (PIs). The amount in which each of these drugs is administered is varied according to the amount of HIV viruses in the body (*Craig, 2004*). Draw a block diagram of a feedback system designed to control the amount of HIV viruses in an infected person. The plant input variables are the amount of RTIs and PIs dispensed. Show blocks representing the controller, the system under control, and the transducers. Label the corresponding variables at the input and output of every block.

23. Hybrid vehicle. The use of hybrid cars is becoming increasingly popular. A hybrid electric vehicle (HEV) combines electric machine(s) with an internal combustion engine (ICE), making it possible (along with other fuel consumption–reducing measures, such as stopping the ICE at traffic lights) to use smaller and more efficient gasoline engines. Thus, the efficiency advantages of the electric drivetrain are obtained, while the energy needed to power the electric motor is stored in the onboard fuel tank and not in a large and heavy battery pack.

There are various ways to arrange the flow of power in hybrid car. In a serial HEV (Figure P1.12), the ICE is not connected to the drive shaft. It drives only the generator, which charges the batteries and/or supplies power to the electric motor(s) through an inverter or a converter.

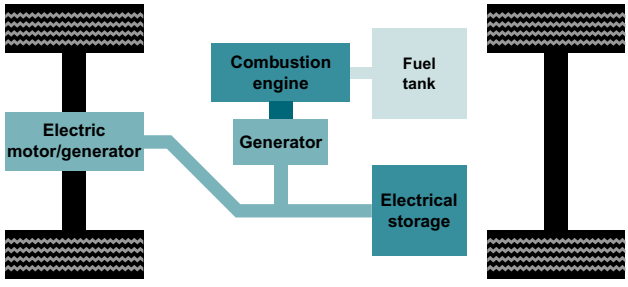


FIGURE P1.12 Serial hybrid-electric vehicle

The HEVs sold today are primarily of the parallel or split-power variety. If the combustion engine can turn the drive wheels as well as the generator, then the vehicle is referred to as a *parallel* hybrid, because both an electric motor and the ICE can drive the vehicle. A parallel hybrid car (Figure P1.13) includes a relatively small battery pack (electrical storage) to put out extra power to the electric motor when fast acceleration is needed. See (*Bosch 5th ed., 2000*), (*Bosch 7th ed., 2007*), (*Edelson, 2008*), (*Anderson, 2009*) for more detailed information about HEV.

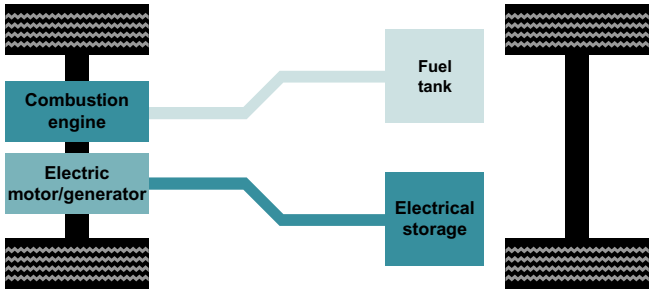


FIGURE P1.13 Parallel hybrid drive

As shown in Figure P1.14, split-power hybrid cars utilize a combination of series and parallel drives (*Bosch, 5th ed., 2007*). These cars use a planetary gear (3) as a split-power transmission to allow some of the ICE power to be applied mechanically to the drive. The other part is converted into electrical energy through the alternator (7) and the inverter (5) to feed the electric motor (downstream of the transmission) and/or to charge the high-voltage

battery (6). Depending upon driving conditions, the ICE, the electric motor, or both propel the vehicle.

1. internal-combustion engine; 2. tank
3. planetary gear; 4. electric motor; 5. inverter;
6. battery; 7. alternator.

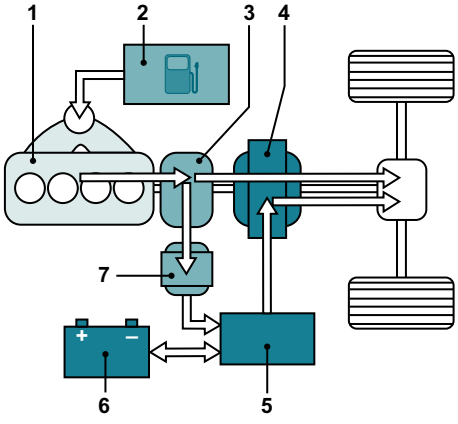


FIGURE P1.14 Split-power hybrid electric vehicle

Draw a functional block diagram for the cruise (speed) control system of:

a. A serial hybrid vehicle, showing its major components, including the speed sensor, electronic control unit (ECU), inverter, electric motor, and vehicle dynamics; as well as all signals, including the desired vehicle speed, actual speed, control command (ECU output), controlled voltage (inverter output), the motive force (provided by the electric motor), and running resistive force[7];

b. A parallel hybrid vehicle, showing its major components, which should include also a block that represents the accelerator, engine, and motor, as well as the signals (including accelerator displacement and combined engine/motor motive force);

c. A split-power HEV, showing its major components and signals, including, in addition to those listed in Parts **a** and **b**, a block representing the planetary gear and its control, which, depending upon driving conditions, would allow the ICE, the electric motor, or both to propel the vehicle, that is, to provide the necessary total motive force.

[7] These include the aerodynamic drag, rolling resistance, and climbing resistance. The aerodynamic drag is a function of car speed, whereas the other two are proportional to car weight.

Cyber Exploration Laboratory

Experiment 1.1

Objective To verify the behavior of closed-loop systems as described in the Chapter 1 Case Study.

Minimum Required Software Packages LabVIEW and the LabVIEW Control Design and Simulation Module. *Note*: While no knowledge of LabVIEW is required for this experiment, see Appendix D to learn more about LabVIEW, which will be pursued in more detail in later chapters.

Prelab

1. From the discussion in the Case Study, describe the effect of the gain of a closed-loop system upon transient response.
2. From the discussion in the Case Study about steady-state error, sketch a graph of a step input superimposed with a step response output and show the steady-state error. Assume any transient response. Repeat for a ramp input and ramp output response. Describe the effect of gain upon the steady-state error.

Lab

1. Launch LabVIEW and open **Find Examples . . .**

2. In the **NI Example Finder** window, open **CDEx Effect of Controller Type.vi,** found by navigating to it thourgh **Toolkits and Modules/Control and Simulation/ Control Design/Time Analysis/CDEx Effect of Controller Type vi.**
3. On the tool bar click the circulating arrows located next to the solid arrow on the left. The program is now running.
4. Move the slider **Controller Gain** and note the effect of high and low gains.
5. Change the controller by clicking the arrows for **Controller Type** and repeat Step 4.

Postlab

1. Correlate the responses found in the experiment with those described in your Prelab. Explore other examples provided in the LabVIEW example folders.

Bibliography

Alternative Drivetrains, July 2005. Available at www.altfuels.org/backgrnd/altdrive.html. Accessed October 13, 2009.

Anderson, S. Field Guide: Hybrid Electric Powertrains, part 4 of 5. *Automotive Design & Production*. Gardner Publication, Inc. Available at http://www.autofieldguide.com/articles/020904.html. Accessed October 13, 2009.

Ayers, J. Taking the Mystery out of Winder Controls. *Power Transmission Design*, April 1988, pp. 27–34.

Bahill, A. T. *Bioengineering: Biomedical, Medical, and Clinical Engineering*. Prentice Hall, Englewood Cliffs, NJ, 1981.

Bechhoefer, J. Feedback for Physicists: A Tutorial Essay on Control. *To Appear in Review of Modern Physics*, July 2005, pp. 42–45. Also available at http://www.sfu.ca/chaos/Publications/papers/RMP_feedback.pdf.

Bennett, S. *A History of Control Engineering, 1800–1930*. Peter Peregrinus, Stevenage, UK, 1979.

Bode, H. W. *Network Analysis and Feedback Amplifier Design*. Van Nostrand, Princeton, NJ, 1945.

Bosch, R. GmbH. *Automotive Electrics and Automotive Electronics*, 5th ed. John Wiley & Sons Ltd., UK, 2007.

Bosch, R. GmbH. *Bosch Automotive Handbook*, 7th ed. John Wiley & Sons Ltd., UK, 2007.

Cannon, R. H., Jr. *Dynamics of Physical Systems*. McGraw-Hill, New York, 1967.

Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

D'Azzo, J. J., and Houpis, C. H. *Feedback Control System Analysis and Synthesis*, 2d ed. McGraw-Hill, New York, 1966.

Doebelin, E. O. *Measurement Systems Application and Design*, 4th ed. McGraw-Hill, New York, 1990.

Dorf, R. C. *Modern Control Systems*, 5th ed. Addison-Wesley, Reading, MA, 1989.

D'Souza, A. F. *Design of Control Systems*. Prentice Hall, Upper Saddle River, NJ, 1988.

Edelson, J., et al. Facing the Challenges of the Current Hybrid Electric Drivetrain. *SMMA Technical Conference of the Motor and Motion Association*. Fall 2008. Available at www.ChorusCars.com.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems*. Addison-Wesley, Reading, MA, 1986.

Heller, H. C., Crawshaw, L. I., and Hammel, H. T. The Thermostat of Vertebrate Animals. *Scientific American*, August 1978, pp. 102–113.

Hogan, B. J. As Motorcycle's Speed Changes, Circuit Adjusts Radio's Volume. *Design News*, 18, August 1988, pp. 118–119.

Hostetter, G. H., Savant, C. J., Jr., and Stefani, R. T. *Design of Feedback Control Systems*, 2d ed. Saunders College Publishing, New York, 1989.

Jenkins, H. E., Kurfess, T. R., and Ludwick, S. J. Determination of a Dynamic Grinding Model. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, June 1997, pp. 289–293.

Klapper, J., and Frankle, J. T. *Phase-Locked and Frequency-Feedback Systems.* Academic Press, New York, 1972.

Lieberman, J., and Breazeal, C. Development of a Wearable Vibrotactile Feedback Suit for Accelerated Human Motor Learning. *2007 IEEE Int. Conf. on Robotics and Automation.* Roma, Italy, April 2007.

Martin, R. H., Jr. *Elementary Differential Equations with Boundary Value Problems.* McGraw-Hill, New York, 1984.

Mayr, O. The Origins of Feedback Control. *Scientific American*, October 1970, pp. 110–118.

Mayr, O. *The Origins of Feedback Control*. MIT Press, Cambridge, MA, 1970.

Meier, R., Nieuwland, J., Zbinden, A. M., and Hacisalihzade, S. S. Fuzzy Logic Control of Blood Pressure during Anesthesia. *IEEE Control Systems*, December 1992, pp. 12–17.

Mott, C., et al. *Modifying the Human Circadian Pacemaker Using Model-Based Predictive Control*. Proceedings of the American Control Conference. Denver, CO, June 2003, pp. 453–458.

Novosad, J. P. *Systems, Modeling, and Decision Making.* Kendall/Hunt, Dubuque, IA, 1982.

Nyquist, H. Regeneration Theory. *Bell System Technical Journal*, January 1932.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. 1–4.

Ogata, K. *Modern Control Engineering*, 2d ed. Prentice Hall, Upper Saddle River, NJ, 1990.

Rockwell International. *Space Shuttle Transportation System*, 1984 (press information).

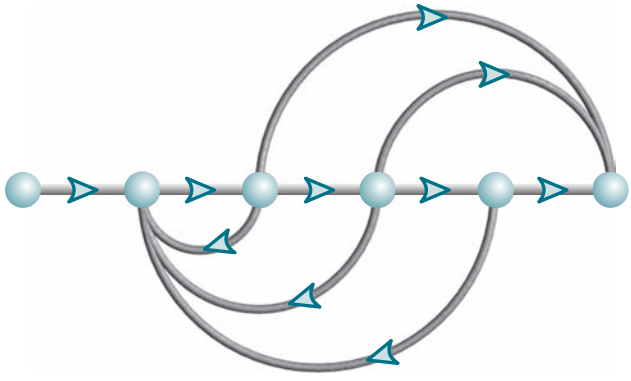
Shaw, D. A., and Turnbull, G. A. Modern Thickness Control for a Generation III Hot Strip Mill. *The International Steel Rolling Conference—The Science & Technology of Flat Rolling*, vol. 1. Association Technique de la Siderurgie Francaise, Deauville, France, 1–3, June 1987.

UNAIDS. *AIDS Epidemic Update*. World Health Organization, December 2005. Available at www.unaids.org/Epi2005/doc/EPIupdate2005_pdf_en/epi-update2005_en.pdf. Accessed January 1, 2006.

Vaughan, N. D., and Gamble, J. B. The Modeling and Simulation of a Proportional Solenoid Valve. *Journal of Dynamic Systems, Measurements, and Control*, vol. 118, March 1996, pp. 120–125.

Modeling in the Frequency Domain

2



Chapter Learning Outcomes

After completing this chapter, the student will be able to:

- Find the Laplace transform of time functions and the inverse Laplace transform (Sections 2.1–2.2)
- Find the transfer function from a differential equation and solve the differential equation using the transfer function (Section 2.3)
- Find the transfer function for linear, time-invariant electrical networks (Section 2.4)
- Find the transfer function for linear, time-invariant translational mechanical systems (Section 2.5)
- Find the transfer function for linear, time-invariant rotational mechanical systems (Section 2.6)
- Find the transfer functions for gear systems with no loss and for gear systems with loss (Section 2.7)
- Find the transfer function for linear, time-invariant electromechanical systems (Section 2.8)
- Produce analogous electrical and mechanical circuits (Section 2.9)
- Linearize a nonlinear system in order to find the transfer function (Sections 2.10–2.11)

Case Study Learning Outcomes

You will be able to demonstrate your knowledge of the chapter objectives with case studies as follows:

- Given the antenna azimuth position control system shown on the front endpapers, you will be able to find the transfer function of each subsystem.
- Given a model of a human leg or a nonlinear electrical circuit, you will be able to linearize the model and then find the transfer function.

2.1 Introduction

In Chapter 1, we discussed the analysis and design sequence that included obtaining the system's schematic and demonstrated this step for a position control system. To obtain a schematic, the control systems engineer must often make many simplifying assumptions in order to keep the ensuing model manageable and still approximate physical reality.

The next step is to develop mathematical models from schematics of physical systems. We will discuss two methods: (1) transfer functions in the frequency domain and (2) state equations in the time domain. These topics are covered in this chapter and in Chapter 3, respectively. As we proceed, we will notice that in every case the first step in developing a mathematical model is to apply the fundamental physical laws of science and engineering. For example, when we model electrical networks, Ohm's law and Kirchhoff's laws, which are basic laws of electric networks, will be applied initially. We will sum voltages in a loop or sum currents at a node. When we study mechanical systems, we will use Newton's laws as the fundamental guiding principles. Here we will sum forces or torques. From these equations we will obtain the relationship between the system's output and input.

In Chapter 1 we saw that a differential equation can describe the relationship between the input and output of a system. The form of the differential equation and its coefficients are a formulation or description of the system. Although the differential equation relates the system to its input and output, it is not a satisfying representation from a system perspective. Looking at Eq. (1.2), a general, nth-order, linear, time-invariant differential equation, we see that the system parameters, which are the coefficients, as well as the output, $c(t)$, and the input, $r(t)$, appear throughout the equation.

We would prefer a mathematical representation such as that shown in Figure 2.1(a), where the input, output, and system are distinct and separate parts. Also, we would like to represent conveniently the interconnection of several subsystems. For example, we would like to represent *cascaded* interconnections, as shown

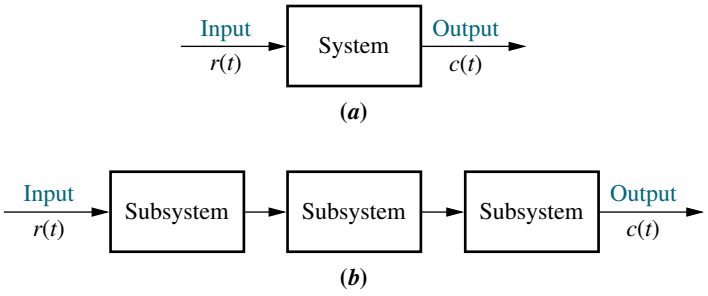


FIGURE 2.1 **a.** Block diagram representation of a system; **b.** block diagram representation of an interconnection of subsystems

Note: The input, $r(t)$, stands for *reference input*. The output, $c(t)$, stands for *controlled variable*.

in Figure 2.1(*b*), where a mathematical function, called a transfer function, is inside each block, and block functions can easily be combined to yield Figure 2.1 (*a*) for ease of analysis and design. This convenience cannot be obtained with the differential equation.

2.2 Laplace Transform Review

A system represented by a differential equation is difficult to model as a block diagram. Thus, we now lay the groundwork for the Laplace transform, with which we can represent the input, output, and system as separate entities. Further, their interrelationship will be simply algebraic. Let us first define the Laplace transform and then show how it simplifies the representation of physical systems (*Nilsson, 1996*).

The Laplace transform is defined as

$$\mathscr{L}[f(t)] = F(s) = \int_{0-}^{\infty} f(t)e^{-st}\,dt \tag{2.1}$$

where $s = \sigma + j\omega$, a complex variable. Thus, knowing $f(t)$ and that the integral in Eq. (2.1) exists, we can find a function, $F(s)$, that is called the *Laplace transform* of $f(t)$.[1]

The notation for the lower limit means that even if $f(t)$ is discontinuous at $t = 0$, we can start the integration prior to the discontinuity as long as the integral converges. Thus, we can find the Laplace transform of impulse functions. This property has distinct advantages when applying the Laplace transform to the solution of differential equations where the initial conditions are discontinuous at $t = 0$. Using differential equations, we have to solve for the initial conditions after the discontinuity knowing the initial conditions before the discontinuity. Using the Laplace transform we need only know the initial conditions before the discontinuity. See *Kailath (1980)* for a more detailed discussion.

The inverse Laplace transform, which allows us to find $f(t)$ given $F(s)$, is

$$\mathscr{L}^{-1}[F(s)] = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}ds = f(t)u(t) \tag{2.2}$$

where

$$\begin{aligned} u(t) &= 1 \quad t > 0 \\ &= 0 \quad t < 0 \end{aligned}$$

is the unit step function. Multiplication of $f(t)$ by $u(t)$ yields a time function that is zero for $t < 0$.

Using Eq. (2.1), it is possible to derive a table relating $f(t)$ to $F(s)$ for specific cases. Table 2.1 shows the results for a representative sample of functions. If we use the tables, we do not have to use Eq. (2.2), which requires complex integration, to find $f(t)$ given $F(s)$.

[1] The Laplace transform exists if the integral of Eq. (2.1) converges. The integral will converge if $\int_{0-}^{\infty} |f(t)|e^{-\sigma_1 t}\,dt < \infty$. If $|f(t)| < Me^{\sigma_2 t}, 0 < t < \infty$, the integral will converge if $\infty > \sigma_1 > \sigma_2$. We call σ_2 the *abscissa of convergence*, and it is the smallest value of σ, where $s = \sigma + j\omega$, for which the integral exists.

TABLE 2.1 Laplace transform table

Item no.	$f(t)$	$F(s)$
1.	$\delta(t)$	1
2.	$u(t)$	$\frac{1}{s}$
3.	$tu(t)$	$\frac{1}{s^2}$
4.	$t^n u(t)$	$\frac{n!}{s^n+1}$
5.	$e^{-at}u(t)$	$\frac{1}{s+a}$
6.	$\sin \omega t u(t)$	$\frac{\omega}{s^2+\omega^2}$
7.	$\cos \omega t u(t)$	$\frac{s}{s^2+\omega^2}$

In the following example we demonstrate the use of Eq. (2.1) to find the Laplace transform of a time function.

Example 2.1

Laplace Transform of a Time Function

PROBLEM: Find the Laplace transform of $f(t) = Ae^{-at}u(t)$.

SOLUTION: Since the time function does not contain an impulse function, we can replace the lower limit of Eq. (2.1) with 0. Hence,

$$F(s) = \int_0^\infty f(t)e^{-st}\,dt = \int_0^\infty Ae^{-at}e^{-st}\,dt = A\int_0^\infty e^{-(s+a)t}\,dt$$

$$= -\frac{A}{s+a}e^{-(s+a)t}\Big|_{t=0}^{\infty} = \frac{A}{s+a} \qquad (2.3)$$

In addition to the Laplace transform table, Table 2.1, we can use Laplace transform theorems, listed in Table 2.2, to assist in transforming between $f(t)$ and $F(s)$. In the next example, we demonstrate the use of the Laplace transform theorems shown in Table 2.2 to find $f(t)$ given $F(s)$.

Example 2.2

Inverse Laplace Transform

PROBLEM: Find the inverse Laplace transform of $F_1(s) = 1/(s+3)^2$.

SOLUTION: For this example we make use of the frequency shift theorem, Item 4 of Table 2.2, and the Laplace transform of $f(t) = tu(t)$, Item 3 of Table 2.1. If the inverse transform of $F(s) = 1/s^2$ is $tu(t)$, the inverse transform of $F(s+a) = 1/(s+a)^2$ is $e^{-at}tu(t)$. Hence, $f_1(t) = e^{-3t}tu(t)$.

TABLE 2.2 Laplace transform theorems

Item no.	Theorem	Name
1.	$\mathscr{L}[f(t)] = F(s) = \int_{0-}^{\infty} f(t)e^{-st}dt$	Definition
2.	$\mathscr{L}[kf(t)] = kF(s)$	Linearity theorem
3.	$\mathscr{L}[f_1(t) + f_2(t)] = F_1(s) + F_2(s)$	Linearity theorem
4.	$\mathscr{L}[e^{-at}f(t)] = F(s+a)$	Frequency shift theorem
5.	$\mathscr{L}[f(t-T)] = e^{-sT}F(s)$	Time shift theorem
6.	$\mathscr{L}[f(at)] = \frac{1}{a}F\left(\frac{s}{a}\right)$	Scaling theorem
7.	$\mathscr{L}\left[\frac{df}{dt}\right] = sF(s) - f(0-)$	Differentiation theorem
8.	$\mathscr{L}\left[\frac{d^2f}{dt^2}\right] = s^2F(s) - sf(0-) - f'(0-)$	Differentiation theorem
9.	$\mathscr{L}\left[\frac{d^nf}{dt^n}\right] = s^nF(s) - \sum_{k=1}^{n} s^{n-k}f^{k-1}(0-)$	Differentiation theorem
10.	$\mathscr{L}\left[\int_{0-}^{t} f(\tau)d\tau\right] = \frac{F(s)}{s}$	Integration theorem
11.	$f(\infty) = \lim_{s\to 0} sF(s)$	Final value theorem[1]
12.	$f(0+) = \lim_{s\to\infty} sF(s)$	Initial value theorem[2]

[1]For this theorem to yield correct finite results, all roots of the denominator of $F(s)$ must have negative real parts, and no more than one can be at the origin.

[2]For this theorem to be valid, $f(t)$ must be continuous or have a step discontinuity at $t = 0$ (that is, no impulses or their derivatives at $t = 0$).

Partial-Fraction Expansion

To find the inverse Laplace transform of a complicated function, we can convert the function to a sum of simpler terms for which we know the Laplace transform of each term. The result is called a *partial-fraction expansion*. If $F_1(s) = N(s)/D(s)$, where the order of $N(s)$ is less than the order of $D(s)$, then a partial-fraction expansion can be made. If the order of $N(s)$ is greater than or equal to the order of $D(s)$, then $N(s)$ must be divided by $D(s)$ successively until the result has a remainder whose numerator is of order less than its denominator. For example, if

$$F_1(s) = \frac{s^3 + 2s^2 + 6s + 7}{s^2 + s + 5} \tag{2.4}$$

we must perform the indicated division until we obtain a remainder whose numerator is of order less than its denominator. Hence,

$$F_1(s) = s + 1 + \frac{2}{s^2 + s + 5} \tag{2.5}$$

Taking the inverse Laplace transform, using Item 1 of Table 2.1, along with the differentiation theorem (Item 7) and the linearity theorem (Item 3 of Table 2.2), we obtain

$$f_1(t) = \frac{d\delta(t)}{dt} + \delta(t) + \mathscr{L}^{-1}\left[\frac{2}{s^2 + s + 5}\right] \tag{2.6}$$

Using partial-fraction expansion, we will be able to expand functions like $F(s) = 2/(s^2 + s + 5)$ into a sum of terms and then find the inverse Laplace transform for each term. We will now consider three cases and show for each case how an $F(s)$ can be expanded into partial fractions.

Case 1. Roots of the Denominator of *F(s)* Are Real and Distinct An example of an $F(s)$ with real and distinct roots in the denominator is

$$F(s) = \frac{2}{(s+1)(s+2)} \tag{2.7}$$

The roots of the denominator are distinct, since each factor is raised only to unity power. We can write the partial-fraction expansion as a sum of terms where each factor of the original denominator forms the denominator of each term, and constants, called *residues*, form the numerators. Hence,

$$F(s) = \frac{2}{(s+1)(s+2)} = \frac{K_1}{(s+1)} + \frac{K_2}{(s+2)} \tag{2.8}$$

To find K_1, we first multiply Eq. (2.8) by $(s+1)$, which isolates K_1. Thus,

$$\frac{2}{(s+2)} = K_1 + \frac{(s+1)K_2}{(s+2)} \tag{2.9}$$

Letting s approach -1 eliminates the last term and yields $K_1 = 2$. Similarly, K_2 can be found by multiplying Eq. (2.8) by $(s+2)$ and then letting s approach -2; hence, $K_2 = -2$.

Each component part of Eq. (2.8) is an $F(s)$ in Table 2.1. Hence, $f(t)$ is the sum of the inverse Laplace transform of each term, or

$$f(t) = (2e^{-t} - 2e^{-2t})u(t) \tag{2.10}$$

In general, then, given an $F(s)$ whose denominator has real and distinct roots, a partial-fraction expansion,

$$\begin{aligned} F(s) = \frac{N(s)}{D(s)} &= \frac{N(s)}{(s+p_1)(s+p_2)\cdots(s+p_m)\cdots(s+p_n)} \\ &= \frac{K_1}{(s+p_1)} + \frac{K_2}{(s+p_2)} + \cdots + \frac{K_m}{(s+p_m)} + \cdots + \frac{K_n}{(s+p_n)} \end{aligned} \tag{2.11}$$

can be made if the order of $N(s)$ is less than the order of $D(s)$. To evaluate each residue, K_i, we multiply Eq. (2.11) by the denominator of the corresponding partial fraction. Thus, if we want to find K_m, we multiply Eq. (2.11) by $(s+p_m)$ and get

$$\begin{aligned} (s+p_m)F(s) &= \frac{(s+p_m)N(s)}{(s+p_1)(s+p_2)\cdots(s+p_m)\cdots(s+p_n)} \\ &= (s+p_m)\frac{K_1}{(s+p_1)} + (s+p_m)\frac{K_2}{(s+p_2)} + \cdots + K_m + \cdots \\ &\quad + (s+p_m)\frac{K_n}{(s+p_n)} \end{aligned} \tag{2.12}$$

If we let s approach $-p_m$, all terms on the right-hand side of Eq. (2.12) go to zero except the term K_m, leaving

$$\left.\frac{\cancel{(s+p_m)}N(s)}{(s+p_1)(s+p_2)\cdots\cancel{(s+p_m)}\cdots(s+p_n)}\right|_{s\to -p_m} = K_m \tag{2.13}$$

The following example demonstrates the use of the partial-fraction expansion to solve a differential equation. We will see that the Laplace transform reduces the task of finding the solution to simple algebra.

Example 2.3

Laplace Transform Solution of a Differential Equation

PROBLEM: Given the following differential equation, solve for $y(t)$ if all initial conditions are zero. Use the Laplace transform.

$$\frac{d^2y}{dt^2} + 12\frac{dy}{dt} + 32y = 32u(t) \tag{2.14}$$

SOLUTION: Substitute the corresponding $F(s)$ for each term in Eq. (2.14), using Item 2 in Table 2.1, Items 7 and 8 in Table 2.2, and the initial conditions of $y(t)$ and $dy(t)/dt$ given by $y(0-) = 0$ and $\dot{y}(0-) = 0$, respectively. Hence, the Laplace transform of Eq. (2.14) is

$$s^2Y(s) + 12sY(s) + 32Y(s) = \frac{32}{s} \tag{2.15}$$

Solving for the response, $Y(s)$, yields

$$Y(s) = \frac{32}{s(s^2 + 12s + 32)} = \frac{32}{s(s+4)(s+8)} \tag{2.16}$$

To solve for $y(t)$, we notice that Eq. (2.16) does not match any of the terms in Table 2.1. Thus, we form the partial-fraction expansion of the right-hand term and match each of the resulting terms with $F(s)$ in Table 2.1. Therefore,

$$Y(s) = \frac{32}{s(s+4)(s+8)} = \frac{K_1}{s} + \frac{K_2}{(s+4)} + \frac{K_3}{(s+8)} \tag{2.17}$$

where, from Eq. (2.13),

$$K_1 = \left.\frac{32}{(s+4)(s+8)}\right|_{s\to 0} = 1 \tag{2.18a}$$

$$K_2 = \left.\frac{32}{s(s+8)}\right|_{s\to -4} = -2 \tag{2.18b}$$

$$K_3 = \left.\frac{32}{s(s+4)}\right|_{s\to -8} = 1 \tag{2.18c}$$

Hence,

$$Y(s) = \frac{1}{s} - \frac{2}{(s+4)} + \frac{1}{(s+8)} \tag{2.19}$$

Since each of the three component parts of Eq. (2.19) is represented as an $F(s)$ in Table 2.1, $y(t)$ is the sum of the inverse Laplace transforms of each term. Hence,

$$y(t) = (1 - 2e^{-4t} + e^{-8t})u(t) \tag{2.20}$$

MATLAB ML

Students who are using MATLAB should now run ch2p1 through ch2p8 in Appendix B. This is your first MATLAB exercise. You will learn how to use MATLAB to (1) represent polynomials, (2) find roots of polynomials, (3) multiply polynomials, and (4) find partial-fraction expansions. Finally, Example 2.3 will be solved using MATLAB.

The $u(t)$ in Eq. (2.20) shows that the response is zero until $t = 0$. Unless otherwise specified, all inputs to systems in the text will not start until $t = 0$. Thus, output responses will also be zero until $t = 0$. For convenience, we will leave off the $u(t)$ notation from now on. Accordingly, we write the output response as

$$y(t) = 1 - 2e^{-4t} + e^{-8t} \tag{2.21}$$

TryIt 2.1

Use the following MATLAB and Control System Toolbox statement to form the linear, time-invariant (LTI) transfer function of Eq. (2.22).

```
F=zpk([],[-1 -2 -2],2)
```

Case 2. Roots of the Denominator of *F(s)* Are Real and Repeated An example of an $F(s)$ with real and repeated roots in the denominator is

$$F(s) = \frac{2}{(s+1)(s+2)^2} \tag{2.22}$$

The roots of $(s+2)^2$ in the denominator are repeated, since the factor is raised to an integer power higher than 1. In this case, the denominator root at -2 is a *multiple root* of *multiplicity* 2.

We can write the partial-fraction expansion as a sum of terms, where each factor of the denominator forms the denominator of each term. In addition, each multiple root generates additional terms consisting of denominator factors of reduced multiplicity. For example, if

$$F(s) = \frac{2}{(s+1)(s+2)^2} = \frac{K_1}{(s+1)} + \frac{K_2}{(s+2)^2} + \frac{K_3}{(s+2)} \tag{2.23}$$

then $K_1 = 2$, which can be found as previously described. K_2 can be isolated by multiplying Eq. (2.23) by $(s+2)^2$, yielding

$$\frac{2}{s+1} = (s+2)^2 \frac{K_1}{(s+1)} + K_2 + (s+2)K_3 \tag{2.24}$$

Letting s approach -2, $K_2 = -2$. To find K_3 we see that if we differentiate Eq. (2.24) with respect to s,

$$\frac{-2}{(s+1)^2} = \frac{(s+2)s}{(s+1)^2} K_1 + K_3 \tag{2.25}$$

TryIt 2.2

Use the following MATLAB statements to help you get Eq. (2.26).

```
numf=2;
denf=poly([-1 -2 -2]);
[k,p,k]=residue...
 (numf,denf)
```

K_3 is isolated and can be found if we let s approach -2. Hence, $K_3 = -2$.

Each component part of Eq. (2.23) is an $F(s)$ in Table 2.1; hence, $f(t)$ is the sum of the inverse Laplace transform of each term, or

$$f(t) = 2e^{-t} - 2te^{-2t} - 2e^{-2t} \tag{2.26}$$

If the denominator root is of higher multiplicity than 2, successive differentiation would isolate each residue in the expansion of the multiple root.

In general, then, given an $F(s)$ whose denominator has real and repeated roots, a partial-fraction expansion,

$$\begin{aligned} F(s) &= \frac{N(s)}{D(s)} \\ &= \frac{N(s)}{(s+p_1)^r(s+p_2)\cdots(s+p_n)} \\ &= \frac{K_1}{(s+p_1)^r} + \frac{K_2}{(s+p_1)^{r-1}} + \cdots + \frac{K_r}{(s+p_1)} \\ &\quad + \frac{K_{r+1}}{(s+p_2)} + \cdots + \frac{K_n}{(s+p_n)} \end{aligned} \tag{2.27}$$

can be made if the order of $N(s)$ is less than the order of $D(s)$ and the repeated roots are of multiplicity r at $-p_1$. To find K_1 through K_r for the roots of multiplicity greater than unity, first multiply Eq. (2.27) by $(s+p_1)^r$ getting $F_1(s)$, which is

$$\begin{aligned} F_1(s) &= (s+p_1)^r F(s) \\ &= \frac{(s+p_1)^r N(s)}{(s+p_1)^r(s+p_2)\cdots(s+p_n)} \\ &= K_1 + (s+p_1)K_2 + (s+p_1)^2K_3 + \cdots + (s+p_1)^{r-1}K_r \\ &\quad + \frac{K_{r+1}(s+p_1)^r}{(s+p_2)} + \cdots + \frac{K_n(s+p_1)^r}{(s+p_n)} \end{aligned} \tag{2.28}$$

Immediately, we can solve for K_1 if we let s approach $-p_1$. We can solve for K_2 if we differentiate Eq. (2.28) with respect to s and then let s approach $-p_1$. Subsequent differentiation will allow us to find K_3 through K_r. The general expression for K_1 through K_r for the multiple roots is

$$K_i = \frac{1}{(i-1)!}\frac{d^{i-1}F_1(s)}{ds^{i-1}}\bigg|_{s\to -p_1} \qquad i = 1,2,\ldots,r; \quad 0! = 1 \tag{2.29}$$

Case 3. Roots of the Denominator of *F(s)* Are Complex or Imaginary An example of $F(s)$ with complex roots in the denominator is

$$F(s) = \frac{3}{s(s^2+2s+5)} \tag{2.30}$$

This function can be expanded in the following form:

$$\frac{3}{s(s^2+2s+5)} = \frac{K_1}{s} + \frac{K_2 s + K_3}{s^2+2s+5} \tag{2.31}$$

K_1 is found in the usual way to be $\frac{3}{5}$. K_2 and K_3 can be found by first multiplying Eq. (2.31) by the lowest common denominator, $s(s^2+2s+5)$, and clearing the fractions. After simplification with $K_1 = \frac{3}{5}$, we obtain

$$3 = \left(K_2 + \frac{3}{5}\right)s^2 + \left(K_3 + \frac{6}{5}\right)s + 3 \tag{2.32}$$

TryIt 2.3

Use the following MATLAB and Control System Toolbox statement to form the LTI transfer function of Eq. (2.30).

```
F=tf([3],[1 2 5 0])
```

Balancing coefficients, $\left(K_2+\frac{3}{5}\right)=0$ and $\left(K_3+\frac{6}{5}\right)=0$. Hence $K_2=-\frac{3}{5}$ and $K_3=-\frac{6}{5}$. Thus,

$$F(s)=\frac{3}{s(s^2+2s+5)}=\frac{3/5}{s}-\frac{3}{5}\,\frac{s+2}{s^2+2s+5} \tag{2.33}$$

The last term can be shown to be the sum of the Laplace transforms of an exponentially damped sine and cosine. Using Item 7 in Table 2.1 and Items 2 and 4 in Table 2.2, we get

$$\mathscr{L}[Ae^{-at}\cos\omega t]=\frac{A(s+a)}{(s+a)^2+\omega^2} \tag{2.34}$$

Similarly,

$$\mathscr{L}[Be^{-at}\sin\omega t]=\frac{B\omega}{(s+a)^2+\omega^2} \tag{2.35}$$

Adding Eqs. (2.34) and (2.35), we get

$$\mathscr{L}[Ae^{-at}\cos\omega t+Be^{-at}\sin\omega t]=\frac{A(s+a)+B\omega}{(s+a)^2+\omega^2} \tag{2.36}$$

TryIt 2.4

Use the following MATLAB and Symbolic Math Toolbox statements to get Eq. (2.38) from Eq. (2.30).

```
syms s
f=ilaplace...
  (3/(s*(s^2+2*s+5)));
pretty(f)
```

We now convert the last term of Eq. (2.33) to the form suggested by Eq. (2.36) by completing the squares in the denominator and adjusting terms in the numerator without changing its value. Hence,

$$F(s)=\frac{3/5}{s}-\frac{3}{5}\frac{(s+1)+(1/2)(2)}{(s+1)^2+2^2} \tag{2.37}$$

Comparing Eq. (2.37) to Table 2.1 and Eq. (2.36), we find

$$f(t)=\frac{3}{5}-\frac{3}{5}e^{-t}\left(\cos 2t+\frac{1}{2}\sin 2t\right) \tag{2.38}$$

In order to visualize the solution, an alternate form of $f(t)$, obtained by trigonometric identities, is preferable. Using the amplitudes of the cos and sin terms, we factor out $\sqrt{1^2+(1/2)^2}$ from the term in parentheses and obtain

$$f(t)=\frac{3}{5}-\frac{3}{5}\sqrt{1^2+(1/2)^2}e^{-t}\left(\frac{1}{\sqrt{1^2+(1/2)^2}}\cos 2t+\frac{1/2}{\sqrt{1^2+(1/2)^2}}\sin 2t\right) \tag{2.39}$$

Letting $1/\sqrt{1^2+(1/2)^2}=\cos\phi$ and $(1/2)/\sqrt{1^2+(1/2)^2}=\sin\phi$,

$$f(t)=\frac{3}{5}-\frac{3}{5}\sqrt{1^2+(1/2)^2}e^{-t}(\cos\phi\cos 2t+\sin\phi\sin 2t) \tag{2.40}$$

or

$$f(t)=0.6-0.671e^{-t}\cos(2t-\phi) \tag{2.41}$$

where $\phi = \arctan 0.5 = 26.57^\circ$. Thus, $f(t)$ is a constant plus an exponentially damped sinusoid.

In general, then, given an $F(s)$ whose denominator has complex or purely imaginary roots, a partial-fraction expansion,

$$F(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{(s+p_1)(s^2+as+b)\cdots} = \frac{K_1}{(s+p_1)} + \frac{(K_2 s + K_3)}{(s^2+as+b)} + \cdots \tag{2.42}$$

can be made if the order of $N(s)$ is less than the order of $D(s)$ p_1 is real, and $(s^2 + as + b)$ has complex or purely imaginary roots. The complex or imaginary roots are expanded with $(K_2 s + K_3)$ terms in the numerator rather than just simply K_i, as in the case of real roots. The K_i's in Eq. (2.42) are found through balancing the coefficients of the equation after clearing fractions. After completing the squares on $(s^2 + as + b)$ and adjusting the numerator, $(K_2 s + K_3)/(s^2 + as + b)$ can be put into the form shown on the right-hand side of Eq. (2.36).

Finally, the case of purely imaginary roots arises if $a = 0$ in Eq. (2.42). The calculations are the same.

Another method that follows the technique used for the partial-fraction expansion of $F(s)$ with real roots in the denominator can be used for complex and imaginary roots. However, the residues of the complex and imaginary roots are themselves complex conjugates. Then, after taking the inverse Laplace transform, the resulting terms can be identified as

$$\frac{e^{j\theta} + e^{-j\theta}}{2} = cos\,\theta \tag{2.43}$$

and

$$\frac{e^{j\theta} - e^{-j\theta}}{2j} = sin\,\theta \tag{2.44}$$

For example, the previous $F(s)$ can also be expanded in partial fractions as

$$F(s) = \frac{3}{s(s^2+2s+5)} = \frac{3}{s(s+1+j2)(s+1-j2)} = \frac{K_1}{s} + \frac{K_2}{s+1+j2} + \frac{K_3}{s+1-j2} \tag{2.45}$$

Finding K_2,

$$K_2 = \left.\frac{3}{s(s+1-j2)}\right|_{s\to -1-j2} = -\frac{3}{20}(2+j1) \tag{2.46}$$

Similarly, K_3 is found to be the complex conjugate of K_2, and K_1 is found as previously described. Hence,

$$F(s) = \frac{3/5}{s} - \frac{3}{20}\left(\frac{2+j1}{s+1+j2} + \frac{2-j1}{s+1-j2}\right) \tag{2.47}$$

from which

$$f(t) = \frac{3}{5} - \frac{3}{20}\left[(2+j1)e^{-(1+j2)t} + (2-j1)e^{-(1-j2)t}\right] = \frac{3}{5} - \frac{3}{20}e^{-t}\left[4\left(\frac{e^{j2t}+e^{-j2t}}{2}\right) + 2\left(\frac{e^{j2t}+e^{-j2t}}{2j}\right)\right] \tag{2.48}$$

TryIt 2.5

Use the following MATLAB statements to help you get Eq. (2.47).

```
numf=3
denf=[1 2 5 0]
[k,p,k]=residue...
  (numf,denf)
```

Using Eqs. (2.43) and (2.44), we get

$$f(t) = \frac{3}{5} - \frac{3}{5}e^{-t}\left(\cos 2t + \frac{1}{2}\sin 2t\right) = 0.6 - 0.671e^{-t}\cos(2t - \phi) \tag{2.49}$$

where $\phi = \arctan 0.5 = 26.57^\circ$.

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch2sp1 and ch2sp2 in Appendix F at www.wiley.com/college/nise. You will learn how to construct symbolic objects and then find the inverse Laplace and Laplace transforms of frequency and time functions, respectively. The examples in Case 2 and Case 3 in this section will be solved using the Symbolic Math Toolbox.

Skill-Assessment Exercise 2.1

PROBLEM: Find the Laplace transform of $f(t) = te^{-5t}$.

ANSWER: $F(s) = 1/(s+5)^2$

The complete solution is at www.wiley.com/college/nise.

Skill-Assessment Exercise 2.2

PROBLEM: Find the inverse Laplace transform of $F(s) = 10/[s(s+2)(s+3)^2]$.

ANSWER: $f(t) = \frac{5}{9} - 5e^{-2t} + \frac{10}{3}te^{-3t} + \frac{40}{9}e^{-3t}$

WileyPLUS WPCS Control Solutions

The complete solution is at www.wiley.com/college/nise.

2.3 The Transfer Function

In the previous section we defined the Laplace transform and its inverse. We presented the idea of the partial-fraction expansion and applied the concepts to the solution of differential equations. We are now ready to formulate the system representation shown in Figure 2.1 by establishing a viable definition for a function that algebraically relates a system's output to its input. This function will allow separation of the input, system, and output into three separate and distinct parts, unlike the differential equation. The function will also allow us to *algebraically* combine mathematical representations of subsystems to yield a total system representation.

Let us begin by writing a general nth-order, linear, time-invariant differential equation,

$$a_n\frac{d^n c(t)}{dt^n} + a_{n-1}\frac{d^{n-1}c(t)}{dt^{n-1}} + \cdots + a_0 c(t) = b_m\frac{d^m r(t)}{dt^m} + b_{m-1}\frac{d^{m-1}r(t)}{dt^{m-1}} + \cdots + b_0 r(t) \tag{2.50}$$

where $c(t)$ is the output, $r(t)$ is the input, and the a_i's, b_i's, and the form of the differential equation represent the system. Taking the Laplace transform of both sides,

$$\begin{aligned} a_n s^n C(s) + a_{n-1} s^{n-1} C(s) + \cdots + a_0 C(s) + &\text{ initial condition} \\ &\text{terms involving } c(t) \\ = b_m s^m R(s) + b_{m-1} s^{m-1} R(s) + \cdots + b_0 R(s) + &\text{ initial condition} \\ &\text{terms involving } r(t) \end{aligned} \tag{2.51}$$

Equation (2.51) is a purely algebraic expression. If we assume that *all initial conditions are zero*, Eq. (2.51) reduces to

$$(a_n s^n + a_{n-1} s^{n-1} + \cdots + a_0) C(s) = (b_m s^m + b_{m-1} s^{m-1} + \cdots + b_0) R(s) \tag{2.52}$$

Now form the ratio of the output transform, $C(s)$, divided by the input transform, $R(s)$:

$$\boxed{\frac{C(s)}{R(s)} = G(s) = \frac{(b_m s^m + b_{m-1} s^{m-1} + \cdots + b_0)}{(a_n s^n + a_{n-1} s^{n-1} + \cdots + a_0)}} \tag{2.53}$$

Notice that Eq. (2.53) separates the output, $C(s)$, the input, $R(s)$, and the system, the ratio of polynomials in s on the right. We call this ratio, $G(s)$, the *transfer function* and evaluate it with *zero initial conditions.*

The transfer function can be represented as a block diagram, as shown in Figure 2.2, with the input on the left, the output on the right, and the system transfer function inside the block. Notice that the denominator of the transfer function is identical to the characteristic polynomial of the differential equation. Also, we can find the output, $C(s)$ by using

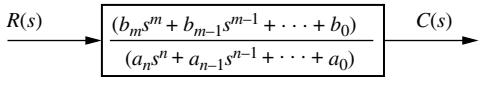


FIGURE 2.2 Block diagram of a transfer function

$$\boxed{C(s) = R(s)G(s)} \tag{2.54}$$

Let us apply the concept of a transfer function to an example and then use the result to find the response of the system.

Example 2.4

Transfer Function for a Differential Equation

PROBLEM: Find the transfer function represented by

$$\frac{dc(t)}{dt} + 2c(t) = r(t) \tag{2.55}$$

SOLUTION: Taking the Laplace transform of both sides, assuming zero initial conditions, we have

$$sC(s) + 2C(s) = R(s) \tag{2.56}$$

The transfer function, $G(s)$, is

$$G(s) = \frac{C(s)}{R(s)} = \frac{1}{s+2} \tag{2.57}$$

MATLAB
ML

Students who are using MATLAB should now run ch2p9 through ch2p12 in Appendix B. You will learn how to use MATLAB to create transfer functions with numerators and denominators in polynomial or factored form. You will also learn how to convert between polynomial and factored forms. Finally, you will learn how to use MATLAB to plot time functions.

Symbolic Math
SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch2sp3 in Appendix F at www.wiley.com/college/nise. You will learn how to use the Symbolic Math Toolbox to simplify the input of complicated transfer functions as well as improve readability. You will learn how to enter a symbolic transfer function and convert it to a linear,time-invariant(LTI) object as presented in Appendix B, ch2p9.

Example 2.5

System Response from the Transfer Function

PROBLEM: Use the result of Example 2.4 to find the response, $c(t)$ to an input, $r(t) = u(t)$, a unit step, assuming zero initial conditions.

SOLUTION: To solve the problem, we use Eq. (2.54), where $G(s) = 1/(s+2)$ as found in Example 2.4. Since $r(t) = u(t)$, $R(s) = 1/s$, from Table 2.1. Since the initial conditions are zero,

$$C(s) = R(s)G(s) = \frac{1}{s(s+2)} \tag{2.58}$$

Expanding by partial fractions, we get

$$C(s) = \frac{1/2}{s} - \frac{1/2}{s+2} \tag{2.59}$$

Finally, taking the inverse Laplace transform of each term yields

$$c(t) = \frac{1}{2} - \frac{1}{2}e^{-2t} \tag{2.60}$$

TryIt 2.6

Use the following MATLAB and Symbolic Math Toolbox statements to help you get Eq. (2.60).

```
syms s
C=1/(s*(s+2))
C=ilaplace(C)
```

TryIt 2.7

Use the following MATLAB statements to plot Eq. (2.60) for t from 0 to 1 sat intervals of 0.01 s.

```
t=0:0.01:1;
plot...
 (t,(1/2-1/2*exp(-2*t)))
```

Skill-Assessment Exercise 2.3

PROBLEM: Find the transfer function, $G(s) = C(s)/R(s)$, corresponding to the differential equation $\frac{d^3c}{dt^3} + 3\frac{d^2c}{dt^2} + 7\frac{dc}{dt} + 5c = \frac{d^2r}{dt^2} + 4\frac{dr}{dt} + 3r$.

ANSWER: $G(s) = \frac{C(s)}{R(s)} = \frac{s^2 + 4s + 3}{s^3 + 3s^2 + 7s + 5}$

The complete solution is at www.wiley.com/college/nise.

Skill-Assessment Exercise 2.4

PROBLEM: Find the differential equation corresponding to the transfer function,

$$G(s) = \frac{2s+1}{s^2+6s+2}$$

ANSWER: $\frac{d^2c}{dt^2} + 6\frac{dc}{dt} + 2c = 2\frac{dr}{dt} + r$

The complete solution is at www.wiley.com/college/nise.

Skill-Assessment Exercise 2.5

PROBLEM: Find the ramp response for a system whose transfer function is

$$G(s) = \frac{s}{(s+4)(s+8)}$$

ANSWER: $c(t) = \frac{1}{32} - \frac{1}{16}e^{-4t} + \frac{1}{32}e^{-8t}$

WileyPLUS
WPCS
Control Solutions

The complete solution is at www.wiley.com/college/nise.

In general, a physical system that can be represented by a linear, time-invariant differential equation can be modeled as a transfer function. The rest of this chapter will be devoted to the task of modeling individual subsystems. We will learn how to represent electrical networks, translational mechanical systems, rotational mechanical systems, and electromechanical systems as transfer functions. As the need arises, the reader can consult the Bibliography at the end of the chapter for discussions of other types of systems, such as pneumatic, hydraulic, and heat-transfer systems (*Cannon, 1967*).

2.4 Electrical Network Transfer Functions

In this section, we formally apply the transfer function to the mathematical modeling of electric circuits including passive networks and operational amplifier circuits. Subsequent sections cover mechanical and electromechanical systems.

Equivalent circuits for the electric networks that we work with first consist of three passive linear components: resistors, capacitors, and inductors.[2] Table 2.3 summarizes the components and the relationships between voltage and current and between voltage and charge under zero initial conditions.

We now combine electrical components into circuits, decide on the input and output, and find the transfer function. Our guiding principles are Kirchhoff's laws. We sum voltages around loops or sum currents at nodes, depending on which technique involves the least effort in algebraic manipulation, and then equate the result to zero. From these relationships we can write the differential equations for the circuit. Then we can take the Laplace transforms of the differential equations and finally solve for the transfer function.

[2] *Passive* means that there is no internal source of energy.

TABLE 2.3 Voltage-current, voltage-charge, and impedance relationships for capacitors, resistors, and inductors

Component	Voltage-current	Current-voltage	Voltage-charge	Impedance $Z(s) = V(s)/I(s)$	Admittance $Y(s) = I(s)/V(s)$
Capacitor	$v(t) = \frac{1}{C}\int_0^t i(\tau)d\tau$	$i(t) = C\frac{dv(t)}{dt}$	$v(t) = \frac{1}{C}q(t)$	$\frac{1}{Cs}$	Cs
Resistor	$v(t) = Ri(t)$	$i(t) = \frac{1}{R}v(t)$	$v(t) = R\frac{dq(t)}{dt}$	R	$\frac{1}{R} = G$
Inductor	$v(t) = L\frac{di(t)}{dt}$	$i(t) = \frac{1}{L}\int_0^t v(\tau)d\tau$	$v(t) = L\frac{d^2q(t)}{dt^2}$	Ls	$\frac{1}{Ls}$

Note: The following set of symbols and units is used throughout this book: $v(t)$ – V (volts), $i(t)$ – A (amps), $q(t)$ – Q (coulombs), C – F (farads), R – Ω (ohms), G – Ω (mhos), L – H (henries).

Simple Circuits via Mesh Analysis

Transfer functions can be obtained using Kirchhoff's voltage law and summing voltages around loops or meshes.[3] We call this method *loop* or *mesh analysis* and demonstrate it in the following example.

Example 2.6

Transfer Function—Single Loop via the Differential Equation

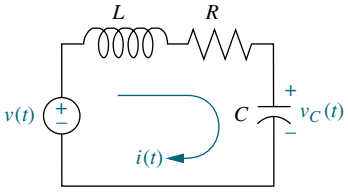


FIGURE 2.3 *RLC* network

PROBLEM: Find the transfer function relating the capacitor voltage, $V_C(s)$, to the input voltage, $V(s)$ in Figure 2.3.

SOLUTION: In any problem, the designer must first decide what the input and output should be. In this network, several variables could have been chosen to be the output—for example, the inductor voltage, the capacitor voltage, the resistor voltage, or the current. The problem statement, however, is clear in this case: We are to treat the capacitor voltage as the output and the applied voltage as the input.

Summing the voltages around the loop, assuming zero initial conditions, yields the integro-differential equation for this network as

$$L\frac{di(t)}{dt} + Ri(t) + \frac{1}{C}\int_0^t i(\tau)d\tau = v(t) \tag{2.61}$$

Changing variables from current to charge using $i(t) = dq(t)/dt$ yields

$$L\frac{d^2q(t)}{dt^2} + R\frac{dq(t)}{dt} + \frac{1}{C}q(t) = v(t) \tag{2.62}$$

From the voltage-charge relationship for a capacitor in Table 2.3,

$$q(t) = Cv_C(t) \tag{2.63}$$

Substituting Eq. (2.63) into Eq. (2.62) yields

$$LC\frac{d^2v_C(t)}{dt^2} + RC\frac{dv_C(t)}{dt} + v_C(t) = v(t) \tag{2.64}$$

[3] A particular loop that resembles the spaces in a screen or fence is called a *mesh*.

Taking the Laplace transform assuming zero initial conditions, rearranging terms, and simplifying yields

$$(LCs^2 + RCs + 1)V_C(s) = V(s) \tag{2.65}$$

Solving for the transfer function, $V_C(s)/V(s)$, we obtain

$$\frac{V_C(s)}{V(s)} = \frac{1/LC}{s^2 + \frac{R}{L}s + \frac{1}{LC}} \tag{2.66}$$

as shown in Figure 2.4.

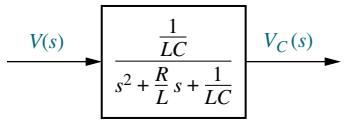


FIGURE 2.4 Block diagram of series *RLC* electrical network

Let us now develop a technique for simplifying the solution for future problems. First, take the Laplace transform of the equations in the voltage-current column of Table 2.3 assuming zero initial conditions.

For the capacitor,

$$V(s) = \frac{1}{Cs}I(s) \tag{2.67}$$

For the resistor,

$$V(s) = RI(s) \tag{2.68}$$

For the inductor,

$$V(s) = LsI(s) \tag{2.69}$$

Now define the following transfer function:

$$\frac{V(s)}{I(s)} = Z(s) \tag{2.70}$$

Notice that this function is similar to the definition of resistance, that is, the ratio of voltage to current. But, unlike resistance, this function is applicable to capacitors and inductors and carries information on the dynamic behavior of the component, since it represents an equivalent differential equation. We call this particular transfer function *impedance*. The impedance for each of the electrical elements is shown in Table 2.3.

Let us now demonstrate how the concept of impedance simplifies the solution for the transfer function. The Laplace transform of Eq. (2.61), assuming zero initial conditions, is

$$\left(Ls + R + \frac{1}{Cs}\right)I(s) = V(s) \tag{2.71}$$

Notice that Eq. (2.71), which is in the form

$$[\text{Sum of impedances}]I(s) = [\text{Sum of applied voltages}] \tag{2.72}$$

suggests the series circuit shown in Figure 2.5. Also notice that the circuit of Figure 2.5 could have been obtained immediately from the circuit of Figure 2.3 simply by replacing each element with its impedance. We call this altered circuit the *transformed circuit*. Finally, notice that the transformed circuit leads immediately to Eq. (2.71) if we add impedances in series as we add resistors in series. Thus, rather than writing the differential equation first and then taking the

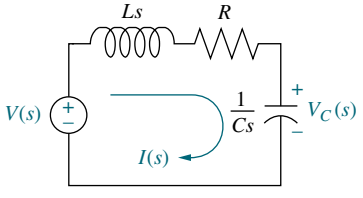


FIGURE 2.5 Laplace-transformed network

Laplace transform, we can draw the transformed circuit and obtain the Laplace transform of the differential equation simply by applying Kirchhoff's voltage law to the transformed circuit. We summarize the steps as follows:

1. Redraw the original network showing all time variables, such as $v(t)$, $i(t)$, and $v_C(t)$, as Laplace transforms $V(s)$, $I(s)$, and $V_C(s)$, respectively.
2. Replace the component values with their impedance values. This replacement is similar to the case of dc circuits, where we represent resistors with their resistance values.

We now redo Example 2.6 using the transform methods just described and bypass the writing of the differential equation.

Example 2.7

Transfer Function—Single Loop via Transform Methods

PROBLEM: Repeat Example 2.6 using mesh analysis and transform methods without writing a differential equation.

SOLUTION: Using Figure 2.5 and writing a mesh equation using the impedances as we would use resistor values in a purely resistive circuit, we obtain

$$\left(Ls + R + \frac{1}{Cs}\right) I(s) = V(s) \tag{2.73}$$

Solving for $I(s)/V(s)$,

$$\frac{I(s)}{V(s)} = \frac{1}{Ls + R + \dfrac{1}{Cs}} \tag{2.74}$$

But the voltage across the capacitor, $V_C(s)$, is the product of the current and the impedance of the capacitor. Thus,

$$V_C(s) = I(s)\frac{1}{Cs} \tag{2.75}$$

Solving Eq. (2.75) for $I(s)$, substituting $I(s)$ into Eq. (2.74), and simplifying yields the same result as Eq. (2.66).

Simple Circuits via Nodal Analysis

Transfer functions also can be obtained using Kirchhoff's current law and summing currents flowing from nodes. We call this method *nodal analysis*. We now demonstrate this principle by redoing Example 2.6 using Kirchhoff's current law and the transform methods just described to bypass writing the differential equation.

Example 2.8

Transfer Function—Single Node via Transform Methods

PROBLEM: Repeat Example 2.6 using nodal analysis and without writing a differential equation.

SOLUTION: The transfer function can be obtained by summing currents flowing out of the node whose voltage is $V_C(s)$ in Figure 2.5. We assume that currents leaving the node are positive and currents entering the node are negative. The currents consist of the current through the capacitor and the current flowing through the series resistor and inductor. From Eq. (2.70), each $I(s) = V(s)/Z(s)$. Hence,

$$\frac{V_C(s)}{I/Cs} + \frac{V_C(s) - V(s)}{R + Ls} = 0 \tag{2.76}$$

where $V_C(s)/(1/Cs)$ is the current flowing out of the node through the capacitor, and $[V_C(s) - V(s)]/(R + Ls)$ is the current flowing out of the node through the series resistor and inductor. Solving Eq. (2.76) for the transfer function, $V_C(s)/V(s)$, we arrive at the same result as Eq. (2.66).

Simple Circuits via Voltage Division

Example 2.6 can be solved directly by using voltage division on the transformed network. We now demonstrate this technique.

Example 2.9

Transfer Function—Single Loop via Voltage Division

PROBLEM: Repeat Example 2.6 using voltage division and the transformed circuit.

SOLUTION: The voltage across the capacitor is some proportion of the input voltage, namely the impedance of the capacitor divided by the sum of the impedances. Thus,

$$V_C(s) = \frac{1/Cs}{\left(Ls + R + \dfrac{1}{Cs}\right)} V(s) \tag{2.77}$$

Solving for the transfer function, $V_C(s)/V(s)$, yields the same result as Eq. (2.66).

Review Examples 2.6 through 2.9. Which method do you think is easiest for this circuit?

The previous example involves a simple, single-loop electrical network. Many electrical networks consist of multiple loops and nodes, and for these circuits we must write and solve simultaneous differential equations in order to find the transfer function, or solve for the output.

Complex Circuits via Mesh Analysis

To solve complex electrical networks—those with multiple loops and nodes—using mesh analysis, we can perform the following steps:

1. Replace passive element values with their impedances.
2. Replace all sources and time variables with their Laplace transform.
3. Assume a transform current and a current direction in each mesh.

4. Write Kirchhoff's voltage law around each mesh.
5. Solve the simultaneous equations for the output.
6. Form the transfer function.

Let us look at an example.

Example 2.10

Transfer Function—Multiple Loops

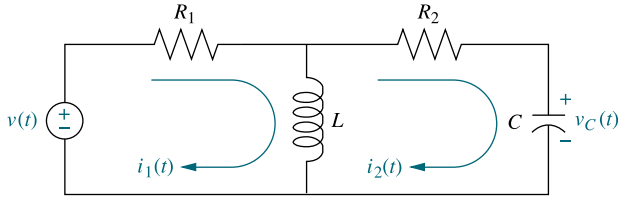
PROBLEM: Given the network of Figure 2.6(a), find the transfer function, $I_2(s)/V(s)$.

SOLUTION: The first step in the solution is to convert the network into Laplace transforms for impedances and circuit variables, assuming zero initial conditions. The result is shown in Figure 2.6(b). The circuit with which we are dealing requires two simultaneous equations to solve for the transfer function. These equations can be found by summing voltages around each mesh through which the assumed currents, $I_1(s)$ and $I_2(s)$, flow. Around Mesh 1, where $I_1(s)$ flows,

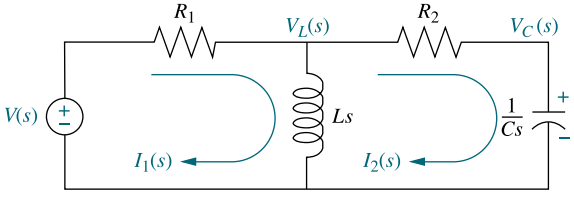
$$R_1 I_1(s) + LsI_1(s) - LsI_2(s) = V(s) \tag{2.78}$$

Around Mesh 2, where $I_2(s)$ flows,

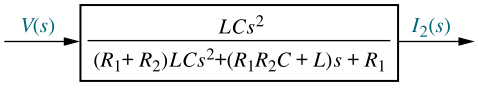
$$LsI_2(s) + R_2 I_2(s) + \frac{1}{Cs} I_2(s) - LsI_1(s) = 0 \tag{2.79}$$



(a)



(b)



(c)

FIGURE 2.6 **a.** Two-loop electrical network; **b.** transformed two-loop electrical network; **c.** block diagram

Combining terms, Eqs. (2.78) and (2.79) become simultaneous equations in $I_1(s)$ and $I_2(s)$:

$$(R_1 + Ls)I_1(s) - LsI_2(s) = V(s) \tag{2.80a}$$

$$-LsI_1(s) + \left(Ls + R_2 + \frac{1}{Cs}\right)I_2(s) = 0 \tag{2.80b}$$

We can use Cramer's rule (or any other method for solving simultaneous equations) to solve Eq. (2.80) for $I_2(s)$.[4] Hence,

$$I_2(s) = \frac{\begin{vmatrix} (R_1 + Ls) & V(s) \\ -Ls & 0 \end{vmatrix}}{\Delta} = \frac{LsV(s)}{\Delta} \tag{2.81}$$

where

$$\Delta = \begin{vmatrix} (R_1 + Ls) & -Ls \\ -Ls & \left(Ls + R_2 + \frac{1}{Cs}\right) \end{vmatrix}$$

Forming the transfer function, $G(s)$, yields

$$G(s) = \frac{I_2(s)}{V(s)} = \frac{Ls}{\Delta} = \frac{LCs^2}{(R_1 + R_2)LCs^2 + (R_1R_2C + L)s + R_1} \tag{2.82}$$

as shown in Figure 2.6(*c*).

We have succeeded in modeling a physical network as a transfer function: The network of Figure 2.6(*a*) is now modeled as the transfer function of Figure 2.6(*c*). Before leaving the example, we notice a pattern first illustrated by Eq. (2.72). The form that Eq. (2.80) take is

$$\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{around Mesh 1}\end{bmatrix} I_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to the}\\ \text{two meshes}\end{bmatrix} I_2(s) = \begin{bmatrix}\text{Sum of applied}\\ \text{voltages around}\\ \text{Mesh 1}\end{bmatrix} \tag{2.83a}$$

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to the}\\ \text{two meshes}\end{bmatrix} I_1(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{around Mesh 2}\end{bmatrix} I_2(s) = \begin{bmatrix}\text{Sum of applied}\\ \text{voltages around}\\ \text{Mesh 2}\end{bmatrix} \tag{2.83b}$$

Recognizing the form will help us write such equations rapidly; for example, mechanical equations of motion (covered in Sections 2.5 and 2.6) have the same form.

Symbolic Math
SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch2sp4 in Appendix F at www.wiley.com/college/nise, where Example 2.10 is solved. You will learn how to use the Symbolic Math Toolbox to solve simultaneous equations using Cramer's rule. Specifically, the Symbolic Math Toolbox will be used to solve for the transfer function in Eq. (2.82) using Eq. (2.80).

[4] See Appendix G (Section G.4) at www.wiley.com/college/nise for Cramer's rule.

Complex Circuits via Nodal Analysis

Often, the easiest way to find the transfer function is to use nodal analysis rather than mesh analysis. The number of simultaneous differential equations that must be written is equal to the number of nodes whose voltage is unknown. In the previous example we wrote simultaneous mesh equations using Kirchhoff's voltage law. For multiple nodes we use Kirchhoff's current law and sum currents flowing from each node. Again, as a convention, currents flowing from the node are assumed to be positive, and currents flowing into the node are assumed to be negative.

Before progressing to an example, let us first define *admittance*, $Y(s)$, as the reciprocal of impedance, or

$$Y(s) = \frac{1}{Z(s)} = \frac{I(s)}{V(s)} \tag{2.84}$$

When writing nodal equations, it can be more convenient to represent circuit elements by their admittance. Admittances for the basic electrical components are shown in Table 2.3. Let us look at an example.

Example 2.11

Transfer Function—Multiple Nodes

PROBLEM: Find the transfer function, $V_C(s)/V(s)$, for the circuit in Figure 2.6(b). Use nodal analysis.

SOLUTION: For this problem, we sum currents at the nodes rather than sum voltages around the meshes. From Figure 2.6(b) the sum of currents flowing from the nodes marked $V_L(s)$ and $V_C(s)$ are, respectively,

$$\frac{V_L(s) - V(s)}{R_1} + \frac{V_L(s)}{Ls} + \frac{V_L(s) - V_C(s)}{R_2} = 0 \tag{2.85a}$$

$$CsV_C(s) + \frac{V_C(s) - V_L(s)}{R_2} = 0 \tag{2.85b}$$

Rearranging and expressing the resistances as conductances,[5] $G_1 = 1/R_1$ and $G_2 = 1/R_2$, we obtain,

$$\left(G_1 + G_2 + \frac{1}{Ls}\right)V_L(s) \qquad - G_2V_C(s) = V(s)G_1 \tag{2.86a}$$

$$-G_2V_L(s) + (G_2 + Cs)V_C(s) = 0 \tag{2.86b}$$

Solving for the transfer function, $V_C(s)/V(s)$, yields

$$\frac{V_C(s)}{V(s)} = \frac{\dfrac{G_1G_2}{C}s}{(G_1 + G_2)s^2 + \dfrac{G_1G_2L + C}{LC}s + \dfrac{G_2}{LC}} \tag{2.87}$$

as shown in Figure 2.7.

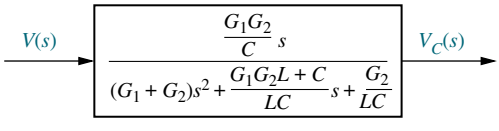


FIGURE 2.7 Block diagram of the network of Figure 2.6

[5] In general, admittance is complex. The real part is called conductance and the imaginary part is called susceptance. But when we take the reciprocal of resistance to obtain the admittance, a purely real quantity results. The reciprocal of resistance is called conductance.

Another way to write node equations is to replace voltage sources by current sources. A voltage source presents a constant voltage to any load; conversely, a current source delivers a constant current to any load. Practically, a current source can be constructed from a voltage source by placing a large resistance in series with the voltage source. Thus, variations in the load do not appreciably change the current, because the current is determined approximately by the large series resistor and the voltage source. Theoretically, we rely on *Norton's theorem*, which states that a voltage source, $V(s)$, in series with an impedance, $Z_s(s)$, can be replaced by a current source, $I(s) = V(s)/Z_s(s)$, in parallel with $Z_s(s)$.

In order to handle multiple-node electrical networks, we can perform the following steps:

1. Replace passive element values with their admittances.
2. Replace all sources and time variables with their Laplace transform.
3. Replace transformed voltage sources with transformed current sources.
4. Write Kirchhoff's current law at each node.
5. Solve the simultaneous equations for the output.
6. Form the transfer function.

Let us look at an example.

Example 2.12

Transfer Function—Multiple Nodes with Current Sources

PROBLEM: For the network of Figure 2.6, find the transfer function, $V_C(s)/V(s)$, using nodal analysis and a transformed circuit with current sources.

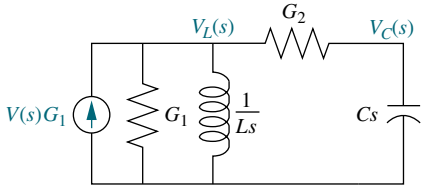


FIGURE 2.8 Transformed network ready for nodal analysis

SOLUTION: Convert all impedances to admittances and all voltage sources in series with an impedance to current sources in parallel with an admittance using Norton's theorem.

Redrawing Figure 2.6(b) to reflect the changes, we obtain Figure 2.8, where $G_1 = 1/R_1$, $G_2 = 1/R_2$, and the node voltages—the voltages across the inductor and the capacitor—have been identified as $V_L(s)$ and $V_C(s)$, respectively. Using the general relationship, $I(s) = Y(s)V(s)$, and summing currents at the node $V_L(s)$,

$$G_1 V_L(s) + \frac{1}{Ls} V_L(s) + G_2[V_L(s) - V_C(s)] = V(s)G_1 \tag{2.88}$$

Summing the currents at the node $V_C(s)$ yields

$$CsV_C(s) + G_2[V_C(s) - V_L(s)] = 0 \tag{2.89}$$

Combining terms, Eqs. (2.88) and (2.89) become simultaneous equations in $V_C(s)$ and $V_L(s)$, which are identical to Eq. (2.86) and lead to the same solution as Eq. (2.87).

An advantage of drawing this circuit lies in the form of Eq. (2.86) and its direct relationship to Figure 2.8, namely

$$\begin{bmatrix}\text{Sum of admittances}\\ \text{connected to Node 1}\end{bmatrix} V_L(s) - \begin{bmatrix}\text{Sum of admittances}\\ \text{common to the two}\\ \text{nodes}\end{bmatrix} V_C(s) = \begin{bmatrix}\text{Sum of applied}\\ \text{currents at Node 1}\end{bmatrix} \tag{2.90a}$$

$$-\begin{bmatrix}\text{Sum of admittances}\\ \text{common to the two}\\ \text{nodes}\end{bmatrix} V_L(s) + \begin{bmatrix}\text{Sum of admittances}\\ \text{connected to Node 2}\end{bmatrix} V_C(s) = \begin{bmatrix}\text{Sum of applied}\\ \text{currents at Node 2}\end{bmatrix} \tag{2.90b}$$

A Problem-Solving Technique

In all of the previous examples, we have seen a repeating pattern in the equations that we can use to our advantage. If we recognize this pattern, we need not write the equations component by component; we can sum impedances around a mesh in the case of mesh equations or sum admittances at a node in the case of node equations. Let us now look at a three-loop electrical network and write the mesh equations by inspection to demonstrate the process.

Example 2.13

Mesh Equations via Inspection

PROBLEM: Write, but do not solve, the mesh equations for the network shown in Figure 2.9.

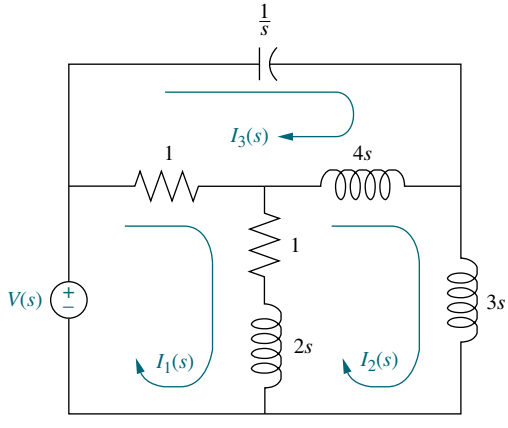


FIGURE 2.9 Three-loop electrical network

SOLUTION: Each of the previous problems has illustrated that the mesh equations and nodal equations have a predictable form. We use that knowledge to solve this three-loop problem. The equation for Mesh 1 will have the following form:

$$\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{around Mesh 1}\end{bmatrix} I_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to}\\ \text{Mesh 1 and}\\ \text{Mesh 2}\end{bmatrix} I_2(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to}\\ \text{Mesh 1 and}\\ \text{Mesh 3}\end{bmatrix} I_3(s) = \begin{bmatrix}\text{Sum of applied}\\ \text{voltages around}\\ \text{Mesh 1}\end{bmatrix} \tag{2.91}$$

Similarly, Meshes 2 and 3, respectively, are

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to}\\ \text{Mesh 1 and}\\ \text{Mesh 2}\end{bmatrix} I_1(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{around Mesh 2}\end{bmatrix} I_2(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to}\\ \text{Mesh 2 and}\\ \text{Mesh 3}\end{bmatrix} I_3(s) = \begin{bmatrix}\text{Sum of appied}\\ \text{voltages around}\\ \text{Mesh 2}\end{bmatrix} \tag{2.92}$$

and

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to}\\ \text{Mesh 1 and}\\ \text{Mesh 3}\end{bmatrix} I_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{common to}\\ \text{Mesh 2 and}\\ \text{Mesh 3}\end{bmatrix} I_2(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{around Mesh 3}\end{bmatrix} I_3(s) = \begin{bmatrix}\text{Sum of applied}\\ \text{voltages around}\\ \text{Mesh 3}\end{bmatrix} \tag{2.93}$$

Substituting the values from Figure 2.9 into Eqs. (2.91) through (2.93) yields

$$+(2s+2)I_1(s) - (2s+1)I_2(s) - I_3(s) = \mathrm{V}(s) \tag{2.94a}$$

$$-(2s+1)I_1(s) + (9s+1)I_2(s) - 4sI_3(s) = 0 \tag{2.94b}$$

$$-I_1(s) - 4sI_2(s) + \left(4s+1+\tfrac{1}{s}\right)I_3(s) = 0 \tag{2.94c}$$

which can be solved simultaneously for any desired transfer function, for example, $I_3(s)/V(s)$.

TryIt 2.8

Use the following MATLAB and Symbolic Math Toolbox statements to help you solve for the electrical currents in Eq. (2.94).

```
syms s I1 I2 I3 V
A=[(2*s+2) -(2*s+1)...
  -1
  -(2*s+1)  (9*s+1)...
  -4*s
  -1 -4*s...
  (4*s+1+1/s)];
B=[I1;I2;I3];
C=[V;0;0];
B=inv(A)*C;
pretty(B)
```

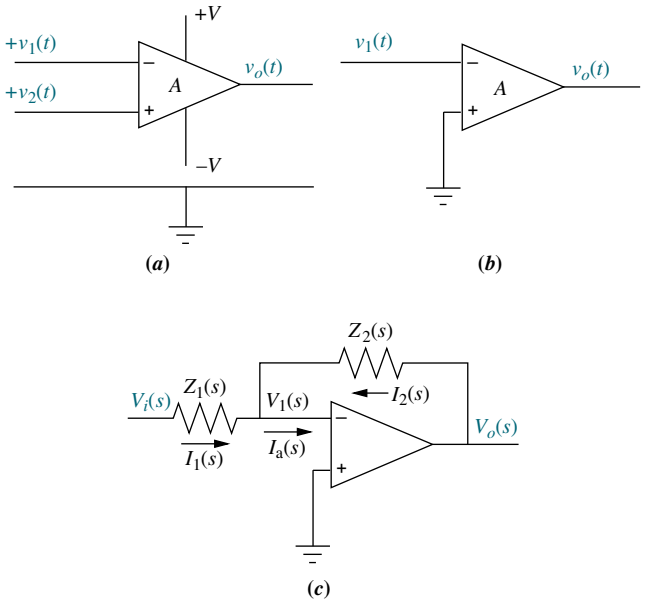


FIGURE 2.10 **a.** Operational amplifier; **b.** schematic for an inverting operational amplifier; **c.** inverting operational amplifier configured for transfer function realization. Typically, the amplifier gain, A, is omitted.

Passive electrical circuits were the topic of discussion up to this point. We now discuss a class of active circuits that can be used to implement transfer functions. These are circuits built around an operational amplifier.

Operational Amplifiers

An *operational amplifier*, pictured in Figure 2.10(a), is an electronic amplifier used as a basic building block to implement transfer functions. It has the following characteristics:

1. Differential input, $V_2(t) - v_1(t)$
2. High input impedance, $Z_i = \infty$ (ideal)
3. Low output impedance, $Z_o = 0$ (ideal)
4. High constant gain amplification, $A = \infty$ (ideal)

The output, $v_o(t)$, is given by

$$v_o(t) = A(v_2(t) - v_1(t)) \tag{2.95}$$

Inverting Operational Amplifier

If $v_2(t)$ is grounded, the amplifier is called an *inverting operational amplifier*, as shown in Figure 2.10(b). For the inverting operational amplifier, we have

$$v_o(t) = -Av_1(t) \tag{2.96}$$

If two impedances are connected to the inverting operational amplifier as shown in Figure 2.10(c), we can derive an interesting result if the amplifier has the characteristics mentioned in the beginning of this subsection. If the input impedance to the amplifier is high, then by Kirchhoff's current law, $I_a(s) = 0$ and $I_1(s) = -I_2(s)$.

Also, since the gain A is large, $v_1(t) \approx 0$. Thus, $I_1(s) = V_i(s)/Z_1(s)$, and $-I_2(s) = -V_o(s)/Z_2(s)$. Equating the two currents, $V_o(s)/Z_2(s) = -V_i(s)/Z_1(s)$, or the transfer function of the inverting operational amplifier configured as shown in Figure 2.10(c) is

$$\boxed{\frac{V_o(s)}{V_i(s)} = -\frac{Z_2(s)}{Z_1(s)}} \tag{2.97}$$

Example 2.14

Transfer Function—Inverting Operational Amplifier Circuit

PROBLEM: Find the transfer function, $V_o(s)/V_i(s)$, for the circuit given in Figure 2.11.

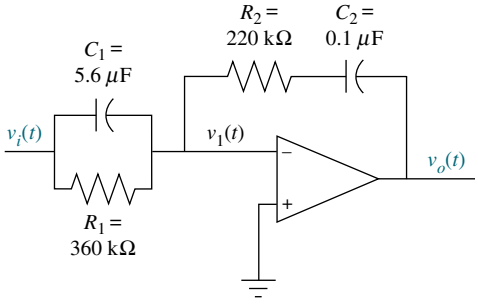


FIGURE 2.11 Inverting operational amplifier circuit for Example 2.14

SOLUTION: The transfer function of the operational amplifier circuit is given by Eq. (2.97). Since the admittances of parallel components add, $Z_1(s)$ is the reciprocal of the sum of the admittances, or

$$Z_1(s) = \frac{1}{C_1 s + \dfrac{1}{R_1}} = \frac{1}{5.6 \times 10^{-6} s + \dfrac{1}{360 \times 10^3}} = \frac{360 \times 10^3}{2.016s + 1} \tag{2.98}$$

For $Z_2(s)$ the impedances add, or

$$Z_2(s) = R_2 + \frac{1}{C_2 s} = 220 \times 10^3 + \frac{10^7}{s} \tag{2.99}$$

Substituting Eqs. (2.98) and (2.99) into Eq. (2.97) and simplifying, we get

$$\frac{V_o(s)}{V_i(s)} = -1.232\frac{s^2 + 45.95s + 22.55}{s} \tag{2.100}$$

The resulting circuit is called a PID controller and can be used to improve the performance of a control system. We explore this possibility further in Chapter 9.

Noninverting Operational Amplifier

Another circuit that can be analyzed for its transfer function is the noninverting operational amplifier circuit shown in Figure 2.12. We now derive the transfer

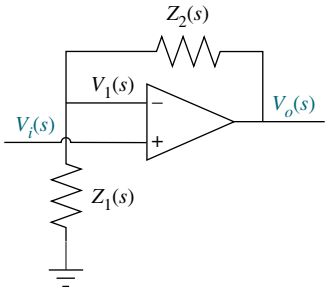


FIGURE 2.12 General noninverting operational amplifier circuit

function. We see that

$$V_o(s) = A(V_i(s) - V_1(s)) \tag{2.101}$$

But, using voltage division,

$$V_1(s) = \frac{Z_1(s)}{Z_1(s) + Z_2(s)} V_o(s) \tag{2.102}$$

Substituting Eq. (2.102) into Eq. (2.101), rearranging, and simplifying, we obtain

$$\frac{V_o(s)}{V_i(s)} = \frac{A}{1 + AZ_1(s)/(Z_1(s) + Z_2(s))} \tag{2.103}$$

For large A, we disregard unity in the denominator and Eq. (2.103) becomes

$$\boxed{\frac{V_o(s)}{V_i(s)} = \frac{Z_1(s) + Z_2(s)}{Z_1(s)}} \tag{2.104}$$

Let us now look at an example.

Example 2.15

Transfer Function—Noninverting Operational Amplifier Circuit

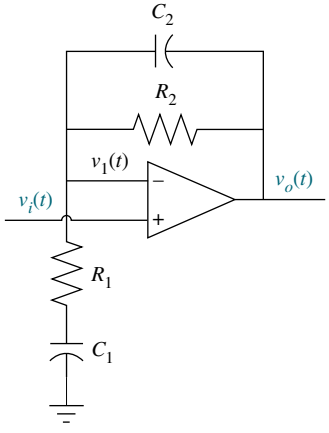


FIGURE 2.13 Noninverting operational amplifier circuit for Example 2.15

PROBLEM: Find the transfer function, $V_o(s)/V_i(s)$, for the circuit given in Figure 2.13.

SOLUTION: We find each of the impedance functions, $Z_1(s)$ and $Z_2(s)$, and then substitute them into Eq. (2.104). Thus,

$$Z_1(s) = R_1 + \frac{1}{C_1 s} \tag{2.105}$$

and

$$Z_2(s) = \frac{R_2(1/C_2 s)}{R_2 + (1/C_2 s)} \tag{2.106}$$

Substituting Eqs. (2.105) and (2.106) into Eq. (2.104) yields

$$\frac{V_o(s)}{V_i(s)} = \frac{C_2C_1R_2R_1s^2 + (C_2R_2 + C_1R_2 + C_1R_1)s + 1}{C_2C_1R_2R_1s^2 + (C_2R_2 + C_1R_1)s + 1} \tag{2.107}$$

Skill-Assessment Exercise 2.6

PROBLEM: Find the transfer function, $G(s) = V_L(s)/V(s)$, for the circuit given in Figure 2.14. Solve the problem two ways—mesh analysis and nodal analysis. Show that the two methods yield the same result.

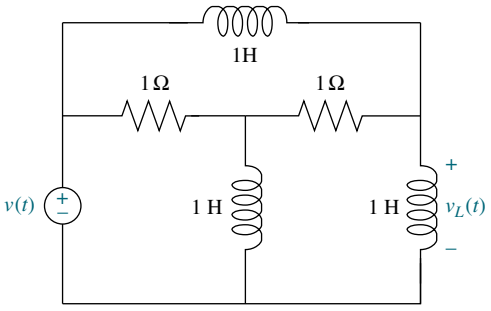


FIGURE 2.14 Electric circuit for Skill-Assessment Exercise 2.6

ANSWER: $V_L(s)/V(s) = (s^2 + 2s + 1)/(s^2 + 5s + 2)$

The complete solution is at www.wiley.com/college/nise.

Skill-Assessment Exercise 2.7

PROBLEM: If $Z_1(s)$ is the impedance of a 10 μF capacitor and $Z_2(s)$ is the impedance of a 100 kΩ resistor, find the transfer function, $G(s) = V_o(s)/V_i(s)$, if these components are used with (a) an inverting operational amplifier and (b) a noninverting amplifier as shown in Figures 2.10(c) and 2.12, respectively.

WileyPLUS
WPCS
Control Solutions

ANSWER: $G(s) = -s$ for an inverting operational amplifier; $G(s) = s + 1$ for a noninverting operational amplifier.

The complete solution is at www.wiley.com/college/nise.

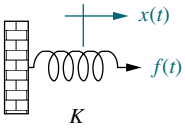
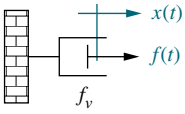
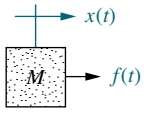
In this section, we found transfer functions for multiple-loop and multiple-node electrical networks, as well as operational amplifier circuits. We developed mesh and nodal equations, noted their form, and wrote them by inspection. In the next section we begin our work with mechanical systems. We will see that many of the concepts applied to electrical networks can also be applied to mechanical systems via analogies—from basic concepts to writing the describing equations by inspection. This revelation will give you the confidence to move beyond this textbook and study systems not covered here, such as hydraulic or pneumatic systems.

2.5 Translational Mechanical System Transfer Functions

We have shown that electrical networks can be modeled by a transfer function, $G(s)$, that algebraically relates the Laplace transform of the output to the Laplace transform of the input. Now we will do the same for mechanical systems. In this section we concentrate on translational mechanical systems. In the next section we extend the concepts to rotational mechanical systems. Notice that the end product, shown in Figure 2.2, will be mathematically indistinguishable from an electrical network. Hence, an electrical network can be interfaced to a mechanical system by cascading their transfer functions, provided that one system is not loaded by the other.[6]

[6] The concept of loading is explained further in Chapter 5.

TABLE 2.4 Force-velocity, force-displacement, and impedance translational relationships for springs, viscous dampers, and mass

Component	Force-velocity	Force-displacement	Impedence $Z_M(s) = F(s)/X(s)$
Spring 	$f(t) = K\int_0^t v(\tau)d\tau$	$f(t) = Kx(t)$	K
Viscous damper 	$f(t) = f_v v(t)$	$f(t) = f_v \frac{dx(t)}{dt}$	$f_v s$
Mass 	$f(t) = M\frac{dv(t)}{dt}$	$f(t) = M\frac{d^2x(t)}{dt^2}$	Ms^2

Note: The following set of symbols and units is used throughout this book: $f(t)$ = N (newtons), $x(t)$ = m (meters), $v(t)$ = m/s (meters/second), K = N/m (newtons/meter), f_v = N-s/m(newton-seconds/meter), M = kg (kilograms = newton-seconds^2/meter).

Mechanical systems parallel electrical networks to such an extent that there are analogies between electrical and mechanical components and variables. Mechanical systems, like electrical networks, have three passive, linear components. Two of them, the spring and the mass, are energy-storage elements; one of them, the viscous damper, dissipates energy. The two energy-storage elements are analogous to the two electrical energy-storage elements, the inductor and capacitor. The energy dissipator is analogous to electrical resistance. Let us take a look at these mechanical elements, which are shown in Table 2.4. In the table, K, f_v, and M are called *spring constant*, *coefficient of viscous friction*, and *mass*, respectively.

We now create analogies between electrical and mechanical systems by comparing Tables 2.3 and 2.4. Comparing the force-velocity column of Table 2.4 to the voltage-current column of Table 2.3, we see that mechanical force is analogous to electrical voltage and mechanical velocity is analogous to electrical current. Comparing the force-displacement column of Table 2.4 with the voltage-charge column of Table 2.3 leads to the analogy between the mechanical displacement and electrical charge. We also see that the spring is analogous to the capacitor, the viscous damper is analogous to the resistor, and the mass is analogous to the inductor. Thus, summing forces written in terms of velocity is analogous to summing voltages written in terms of current, and the resulting mechanical differential equations are analogous to mesh equations. If the forces are written in terms of displacement, the resulting mechanical equations resemble, but are not analogous to, the mesh equations. We, however, will use this model for mechanical systems so that we can write equations directly in terms of displacement.

Another analogy can be drawn by comparing the force-velocity column of Table 2.4 to the current-voltage column of Table 2.3 in reverse order. Here the analogy is between force and current and between velocity and voltage. Also, the

spring is analogous to the inductor, the viscous damper is analogous to the resistor, and the mass is analogous to the capacitor. Thus, summing forces written in terms of velocity is analogous to summing currents written in terms of voltage and the resulting mechanical differential equations are analogous to nodal equations. We will discuss these analogies in more detail in Section 2.9.

We are now ready to find transfer functions for translational mechanical systems. Our first example, shown in Figure 2.15(a), is similar to the simple RLC network of Example 2.6 (see Figure 2.3). The mechanical system requires just one differential equation, called the *equation of motion*, to describe it. We will begin by assuming a positive direction of motion, for example, to the right. This assumed positive direction of motion is similar to assuming a current direction in an electrical loop. Using our assumed direction of positive motion, we first draw a free-body diagram, placing on the body all forces that act on the body either in the direction of motion or opposite to it. Next we use Newton's law to form a differential equation of motion by summing the forces and setting the sum equal to zero. Finally, assuming zero initial conditions, we take the Laplace transform of the differential equation, separate the variables, and arrive at the transfer function. An example follows.

Example 2.16

Transfer Function—One Equation of Motion

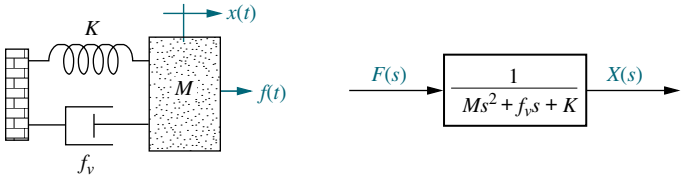


FIGURE 2.15 **a.** Mass, spring, and damper system; **b.** block diagram

PROBLEM: Find the transfer function, $X(s)/F(s)$, for the system of Figure 2.15(a).

SOLUTION: Begin the solution by drawing the free-body diagram shown in Figure 2.16(a). Place on the mass all forces felt by the mass. We assume the mass is traveling toward the right. Thus, only the applied force points to the right; all other forces impede the motion and act to oppose it. Hence, the spring, viscous damper, and the force due to acceleration point to the left.

We now write the differential equation of motion using Newton's law to sum to zero all of the forces shown on the mass in Figure 2.16(a):

$$M\frac{d^2x(t)}{dt^2} + f_v\frac{dx(t)}{dt} + Kx(t) = f(t) \tag{2.108}$$

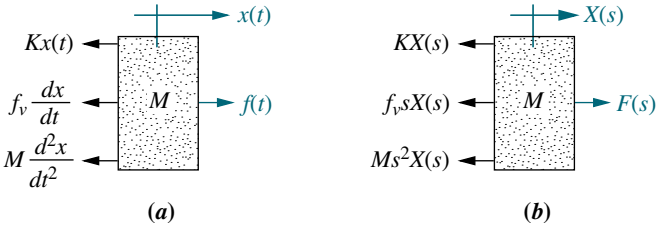


FIGURE 2.16 **a.** Free-body diagram of mass, spring, and damper system; **b.** transformed free-body diagram

Taking the Laplace transform, assuming zero initial conditions,

$$Ms^2X(s) + f_v sX(s) + KX(s) = F(s) \tag{2.109}$$

or

$$(Ms^2 + f_v s + K)X(s) = F(s) \tag{2.110}$$

Solving for the transfer function yields

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{Ms^2 + f_v s + K} \tag{2.111}$$

which is represented in Figure 2.15(b).

Now can we parallel our work with electrical networks by circumventing the writing of differential equations and by defining impedances for mechanical components? If so, we can apply to mechanical systems the problem-solving techniques learned in the previous section. Taking the Laplace transform of the force-displacement column in Table 2.4, we obtain for the spring,

$$\boxed{F(s) = KX(s)} \tag{2.112}$$

for the viscous damper,

$$\boxed{F(s) = f_v sX(s)} \tag{2.113}$$

and for the mass,

$$\boxed{F(s) = Ms^2X(s)} \tag{2.114}$$

If we define impedance for mechanical components as

$$\boxed{Z_M(s) = \frac{F(s)}{X(s)}} \tag{2.115}$$

and apply the definition to Eqs. (2.112) through (2.114), we arrive at the impedances of each component as summarized in Table 2.4 (*Raven, 1995*).[7]

Replacing each force in Figure 2.16(a) by its Laplace transform, which is in the format

$$\boxed{F(s) = Z_M(s)X(s)} \tag{2.116}$$

we obtain Figure 2.16(b), from which we could have obtained Eq. (2.109) immediately without writing the differential equation. From now on we use this approach.

[7] Notice that the impedance column of Table 2.4 is not a direct analogy to the impedance column of Table 2.3, since the denominator of Eq. (2.115) is displacement. A direct analogy could be derived by defining mechanical impedance in terms of velocity as $F(s)/V(s)$. We chose Eq. (2.115) as a convenient definition for writing the equations of motion in terms of displacement, rather than velocity. The alternative, however, is available.

Finally, notice that Eq. (2.110) is of the form

$$[\text{Sum of impedances}]X(s) = [\text{Sum of applied forces}] \tag{2.117}$$

which is similar, but not analogous, to a mesh equation (see footnote 7).

Many mechanical systems are similar to multiple-loop and multiple-node electrical networks, where more than one simultaneous differential equation is required to describe the system. In mechanical systems, the number of equations of motion required is equal to the number of *linearly independent* motions. Linear independence implies that a point of motion in a system can still move if all other points of motion are held still. Another name for the number of linearly independent motions is the number of *degrees of freedom.* This discussion is not meant to imply that these motions are not coupled to one another; in general, they are. For example, in a two-loop electrical network, each loop current depends on the other loop current, but if we open-circuit just one of the loops, the other current can still exist if there is a voltage source in that loop. Similarly, in a mechanical system with two degrees of freedom, one point of motion can be held still while the other point of motion moves under the influence of an applied force.

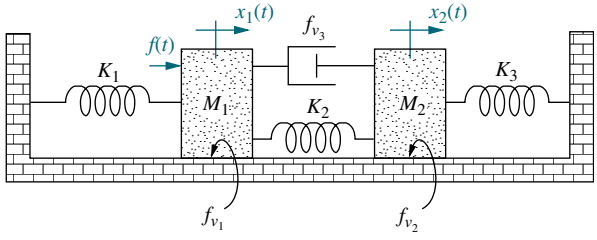
In order to work such a problem, we draw the free-body diagram for each point of motion and then use superposition. For each free-body diagram we begin by holding all other points of motion still and finding the forces acting on the body due only to its own motion. Then we hold the body still and activate the other points of motion one at a time, placing on the original body the forces created by the adjacent motion.

Using Newton's law, we sum the forces on each body and set the sum to zero. The result is a system of simultaneous equations of motion. As Laplace transforms, these equations are then solved for the output variable of interest in terms of the input variable from which the transfer function is evaluated. Example 2.17 demonstrates this problem-solving technique.

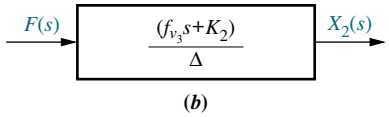
Example 2.17

Transfer Function—Two Degrees of Freedom

PROBLEM: Find the transfer function, $X_2(s)/F(s)$, for the system of Figure 2.17(*a*).



(*a*)



(*b*)

FIGURE 2.17 **a.** Two-degrees-of-freedom translational mechanical system;[8] **b.** block diagram

Virtual Experiment 2.1
Automobile Suspension

Put theory into practice exploring the dynamics of another two degree of freedom system—an automobile suspension system driving over a bumpy road demonstrated with the Quanser Active Suspension System modeled in LabVIEW.



Virtual experiments are found on WileyPLUS.

[8] Friction shown here and throughout the book, unless otherwise indicated, is viscous friction. Thus, f_{v1} and f_{v2} are not Coulomb friction, but arise because of a viscous interface.

SOLUTION: The system has two degrees of freedom, since each mass can be moved in the horizontal direction while the other is held still. Thus, two simultaneous equations of motion will be required to describe the system. The two equations come from free-body diagrams of each mass. Superposition is used to draw the free-body diagrams. For example, the forces on M_1 are due to (1) its own motion and (2) the motion of M_2 transmitted to M_1 through the system. We will consider these two sources separately.

If we hold M_2 still and move M_1 to the right, we see the forces shown in Figure 2.18(a). If we hold M_1 still and move M_2 to the right, we see the forces shown in Figure 2.18(b). The total force on M_1 is the superposition, or sum, of the forces just discussed. This result is shown in Figure 2.18(c). For M_2, we proceed in a similar fashion: First we move M_2 to the right while holding M_1 still; then we move M_1 to the right and hold M_2 still. For each case we evaluate the forces on M_2. The results appear in Figure 2.19.

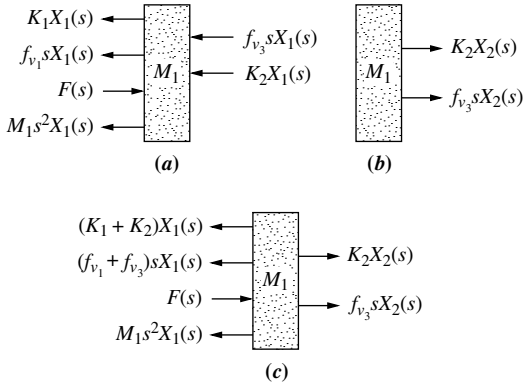


FIGURE 2.18 **a.** Forces on M_1 due only to motion of M_1; **b.** forces on M_1 due only to motion of M_2; **c.** all forces on M_1

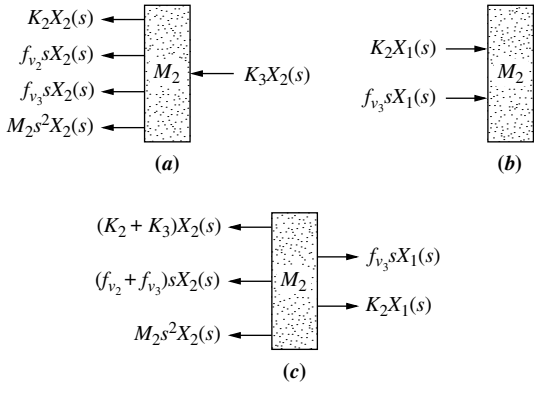


FIGURE 2.19 **a.** Forces on M_2 due only to motion of M_2; **b.** forces on M_2 due only to motion of M_1; **c.** all forces on M_2

The Laplace transform of the equations of motion can now be written from Figures 2.18(c) and 2.19(c) as

$$\left[M_1s^2(f_{v_1} + f_{v_3})s + (K_1 + K_2)\right]X_1(s) - (f_{v_3}s + K_2)X_2(s) = F(s) \qquad (2.118a)$$

$$-(f_{v_3}s + K_2)X_1(s) + \left[M_2s^2 + (f_{v_2} + f_{v_3})s + (K_2 + K_3)\right]X_2(s) = 0 \qquad (2.118b)$$

From this, the transfer function, $X_2(s)/F(s)$, is

$$\frac{X_2(s)}{F(s)} = G(s) = \frac{(f_{v_3}s + K_2)}{\Delta} \tag{2.119}$$

as shown in Figure 2.17(b) where

$$\Delta = \begin{vmatrix} [M_1s^2 + (f_{v_1} + f_{v_3})s + (K_1 + K_2)] & -(f_{v_3}s + K_2) \\ -(f_{v_3}s + K_2) & [M_2s^2 + (f_{v_2} + f_{v_3})s + (K_2 + K_3)] \end{vmatrix}$$

Notice again, in Eq. (2.118), that the form of the equations is similar to electrical mesh equations:

$$\begin{bmatrix} \text{Sum of impedances connected to the motion at } x_1 \end{bmatrix} X_1(s) - \begin{bmatrix} \text{Sum of impedances between } x_1 \text{ and } x_2 \end{bmatrix} X_2(s) = \begin{bmatrix} \text{Sum of applied forces at } x_1 \end{bmatrix} \tag{2.120a}$$

$$-\begin{bmatrix} \text{Sum of impedances between } x_1 \text{ and } x_2 \end{bmatrix} X_1(s) + \begin{bmatrix} \text{Sum of impedances connected to the motion at } x_2 \end{bmatrix} X_2(s) = \begin{bmatrix} \text{Sum of applied forces at } x_2 \end{bmatrix} \tag{2.120b}$$

The pattern shown in Eq. (2.120) should now be familiar to us. Let us use the concept to write the equations of motion of a three-degrees-of-freedom mechanical network by inspection, without drawing the free-body diagram.

Example 2.18

Equations of Motion by Inspection

PROBLEM: Write, but do not solve, the equations of motion for the mechanical network of Figure 2.20.

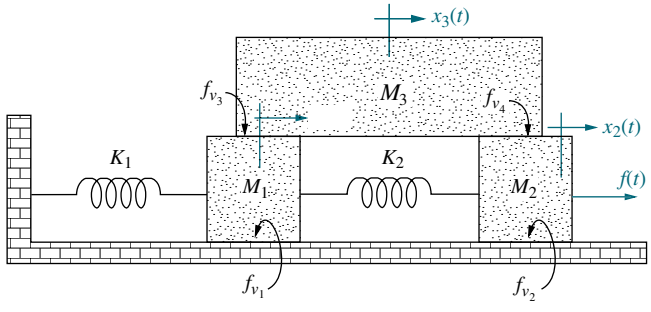


FIGURE 2.20 Three-degrees-of-freedom translational mechanical system

SOLUTION: The system has three degrees of freedom, since each of the three masses can be moved independently while the others are held still. The form of the equations will be similar to electrical mesh equations. For M_1,

$$\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{connected}\\ \text{to the motion}\\ \text{at } x_1\end{bmatrix} X_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ x_1 \text{ and } x_2\end{bmatrix} X_2(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ x_1 \text{ and } x_3\end{bmatrix} X_3(s) = \begin{bmatrix}\text{Sum of}\\ \text{applied forces}\\ \text{at } x_1\end{bmatrix} \tag{2.121}$$

Similarly, for M_2 and M_3, respectively,

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ x_1 \text{ and } x_2\end{bmatrix} X_1(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{connected}\\ \text{to the motion}\\ \text{at } x_2\end{bmatrix} X_2(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ x_2 \text{ and } x_3\end{bmatrix} X_3(s) = \begin{bmatrix}\text{Sum of}\\ \text{applied forces}\\ \text{at } x_2\end{bmatrix} \tag{2.122}$$

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ x_1 \text{ and } x_3\end{bmatrix} X_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ x_2 \text{ and } x_3\end{bmatrix} X_2(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{connected}\\ \text{to the motion}\\ \text{at } x_3\end{bmatrix} X_3(s) = \begin{bmatrix}\text{Sum of}\\ \text{applied forces}\\ \text{at } x_3\end{bmatrix} \tag{2.123}$$

M_1 has two springs, two viscous dampers, and mass associated with its motion. There is one spring between M_1 and M_2 and one viscous damper between M_1 and M_3. Thus, using Eq. (2.121),

$$\left[M_1 s^2 + (f_{v_1} + f_{v_3})s + (K_1 + K_2)\right] X_1(s) - K_2 X_2(s) - f_{v_3} s X_3(s) = 0 \tag{2.124}$$

Similarly, using Eq. (2.122) for M_2,

$$-K_2 X_1(s) + \left[M_2 s^2 + (f_{v_2} + f_{v_4})s + K_2\right] X_2(s) - f_{v_4} s X_3(s) = F(s) \tag{2.125}$$

and using Eq. (2.123) for M_3,

$$-f_{v_3}sX_1(s) - f_{v_4}sX_2(s) + \left[M_3s^2 + (f_{v_3} + f_{v_4})s\right]X_3(s) = 0 \qquad (2.126)$$

Equations (2.124) through (2.126) are the equations of motion. We can solve them for any displacement, $X_1(s)$, $X_2(s)$, or $X_3(s)$, or transfer function.

Skill-Assessment Exercise 2.8

PROBLEM: Find the transfer function, $G(s) = X_2(s)/F(s)$, for the translational mechanical system shown in Figure 2.21.

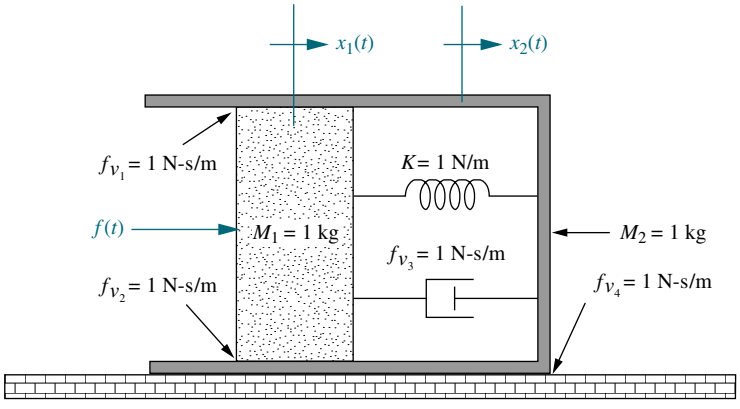


FIGURE 2.21 Translational mechanical system for Skill-Assessment Exercise 2.8

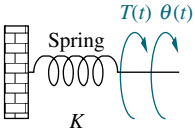
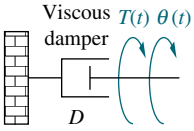
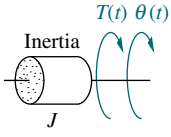
ANSWER: $G(s) = \dfrac{3s+1}{s(s^3+7s^2+5s+1)}$

The complete solution is at www.wiley.com/college/nise.

2.6 Rotational Mechanical System Transfer Functions

Having covered electrical and translational mechanical systems, we now move on to consider rotational mechanical systems. Rotational mechanical systems are handled the same way as translational mechanical systems, except that torque replaces force and angular displacement replaces translational displacement. The mechanical components for rotational systems are the same as those for translational systems, except that the components undergo rotation instead of translation. Table 2.5 shows the components along with the relationships between torque and angular velocity, as well as angular displacement. Notice that the symbols for the

TABLE 2.5 Torque-angular velocity, torque-angular displacement, and impedance rotational relationships for springs, viscous dampers, and inertia

Component	Torque-angular velocity	Torque-angular displacement	Impedence $Z_M(s) = T(s)/\theta(s)$
	$T(t) = K\int_0^t \omega(\tau)d\tau$	$T(t) = K\theta(t)$	K
	$T(t) = D\omega(t)$	$T(t) = D\dfrac{d\theta(t)}{dt}$	Ds
	$T(t) = J\dfrac{d\omega(t)}{dt}$	$T(t) = J\dfrac{d^2\theta(t)}{dt^2}$	Js^2

Note: The following set of symbols and units is used throughout this book: $T(t)$ – N-m (newton-meters), $\theta(t)$ – rad(radians), $\omega(t)$ – rad/s(radians/second), K – N-m/rad(newton- meters/radian), D – N-m-s/rad (newton- meters-seconds/radian). J – kg-m^2(kilograms-meters^2 – newton-meters-seconds^2/radian).

components look the same as translational symbols, but they are undergoing rotation and not translation.

Also notice that the term associated with the mass is replaced by inertia. The values of K, D, and J are called *spring constant*, *coefficient of viscous friction*, and *moment of inertia*, respectively. The impedances of the mechanical components are also summarized in the last column of Table 2.5. The values can be found by taking the Laplace transform, assuming zero initial conditions, of the torque-angular displacement column of Table 2.5.

The concept of degrees of freedom carries over to rotational systems, except that we test a point of motion by *rotating* it while holding still all other points of motion. The number of points of motion that can be rotated while all others are held still equals the number of equations of motion required to describe the system.

Writing the equations of motion for rotational systems is similar to writing them for translational systems; the only difference is that the free-body diagram consists of torques rather than forces. We obtain these torques using superposition. First, we rotate a body while holding all other points still and place on its free-body diagram all torques due to the body's own motion. Then, holding the body still, we rotate adjacent points of motion one at a time and add the torques due to the adjacent motion to the free-body diagram. The process is repeated for each point of motion. For each free-body diagram, these torques are summed and set equal to zero to form the equations of motion.

Two examples will demonstrate the solution of rotational systems. The first one uses free-body diagrams; the second uses the concept of impedances to write the equations of motion by inspection.

Example 2.19

Transfer Function—Two Equations of Motion

PROBLEM: Find the transfer function, $\theta_2(s)/T(s)$, for the rotational system shown in Figure 2.22(a). The rod is supported by bearings at either end and is undergoing torsion. A torque is applied at the left, and the displacement is measured at the right.

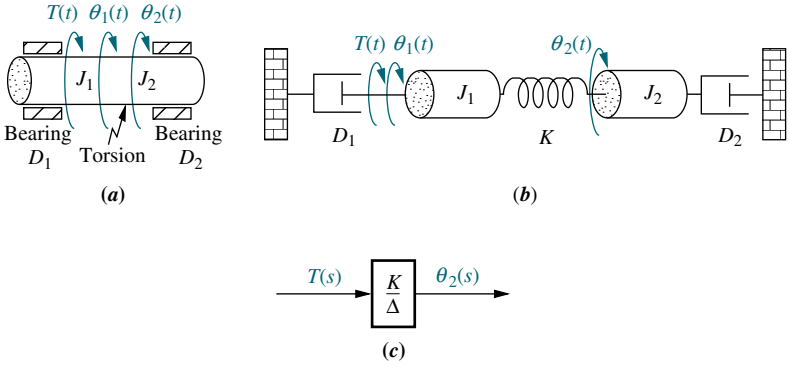


FIGURE 2.22 **a.** Physical system; **b.** schematic; **c.** block diagram

SOLUTION: First, obtain the schematic from the physical system. Even though torsion occurs throughout the rod in Figure 2.22(a),[9] we approximate the system by assuming that the torsion acts like a spring concentrated at one particular point in the rod, with an inertia J_1 to the left and an inertia J_2 to the right.[10] We also assume that the damping inside the flexible shaft is negligible. The schematic is shown in Figure 2.22(b). There are two degrees of freedom, since each inertia can be rotated while the other is held still. Hence, it will take two simultaneous equations to solve the system.

Next, draw a free-body diagram of J_1, using superposition. Figure 2.23(a) shows the torques on J_1 if J_2 is held still and J_1 rotated. Figure 2.23(b) shows the torques on J_1 if J_1 is held still and J_2 rotated. Finally, the sum of Figures 2.23(a) and 2.23(b) is shown in Figure 2.23(c), the final free-body diagram for J_1. The same process is repeated in Figure 2.24 for J_2.

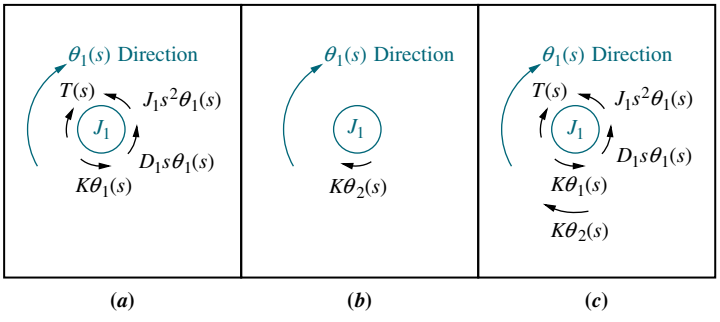


FIGURE 2.23 **a.** Torques on J_1 due only to the motion of J_1; **b.** torques on J_1 due only to the motion of J_2; **c.** final free-body diagram for J_1

[9] In this case the parameter is referred to as a *distributed* parameter.

[10] The parameter is now referred to as a *lumped* parameter.

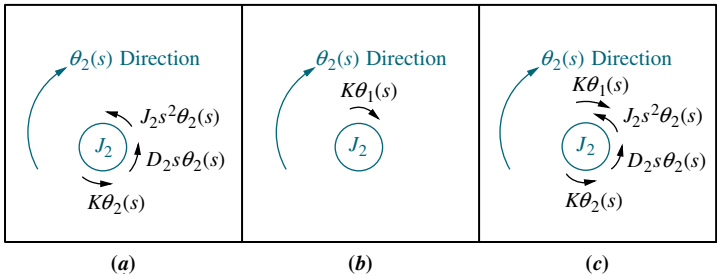


FIGURE 2.24 **a.** Torques on J_2 due only to the motion of J_2; **b.** torques on J_2 due only to the motion of J_1; **c.** final free-body diagram for J_2

Summing torques respectively from Figures 2.23(c) and 2.24(c) we obtain the equations of motion,

$$(J_1 s^2 + D_1 s + K)\theta_1(s) \qquad - K\theta_2(s) = T(s) \tag{2.127a}$$

$$-K\theta_1(s) + (J_2 s^2 + D_2 s + K)\theta_2(s) = 0 \tag{2.127b}$$

from which the required transfer function is found to be

$$\frac{\theta_2(s)}{T(s)} = \frac{K}{\Delta} \tag{2.128}$$

as shown in Figure 2.22(c), where

$$\Delta = \begin{vmatrix} (J_1 s^2 + D_1 s + K) & -K \\ -K & (J_2 s^2 + D_2 s + K) \end{vmatrix}$$

Notice that Eq. (2.127) have that now well-known form

$$\begin{bmatrix} \text{Sum of impedances connected to the motion at } \theta_1 \end{bmatrix} \theta_1(s) - \begin{bmatrix} \text{Sum of impedances between } \theta_1 \text{ and } \theta_2 \end{bmatrix} \theta_2(s) = \begin{bmatrix} \text{Sum of applied torques at } \theta_1 \end{bmatrix} \tag{2.129a}$$

$$- \begin{bmatrix} \text{Sum of impedances between } \theta_1 \text{ and } \theta_2 \end{bmatrix} \theta_1(s) + \begin{bmatrix} \text{Sum of impedances connected to the motion at } \theta_2 \end{bmatrix} \theta_2(s) = \begin{bmatrix} \text{Sum of applied torques at } \theta_2 \end{bmatrix} \tag{2.129b}$$

TryIt 2.9

Use the following MATLAB and Symbolic Math Toolbox statements to help you get Eq. (2.128).

```
syms s J1 D1 K T J2 D2...
  theta1 theta2
A=[(J1*s^2+D1*s+K)  -K
    -K (J2*s^2+D2*s+K)];
B=[theta1
    theta2];
C=[T
    0];
B=inv(A)*C;
theta2=B(2);
'theta2'
pretty(theta2)
```

Example 2.20

Equations of Motion By Inspection

PROBLEM: Write, but do not solve, the Laplace transform of the equations of motion for the system shown in Figure 2.25.

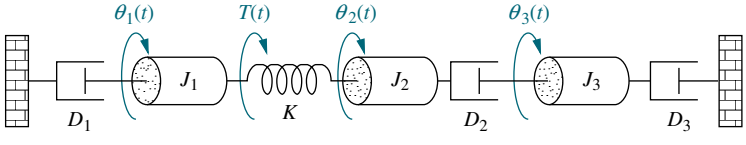


FIGURE 2.25 Three-degrees-of-freedom rotational system

SOLUTION: The equations will take on the following form, similar to electrical mesh equations:

$$\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{connected}\\ \text{to the motion}\\ \text{at } \theta_1\end{bmatrix}\theta_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ \theta_1 \text{ and } \theta_2\end{bmatrix}\theta_2(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ \theta_1 \text{ and } \theta_3\end{bmatrix}\theta_3(s) = \begin{bmatrix}\text{Sum of}\\ \text{applied torques}\\ \text{at } \theta_1\end{bmatrix} \tag{2.130a}$$

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ \theta_1 \text{ and } \theta_2\end{bmatrix}\theta_1(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{connected}\\ \text{to the motion}\\ \text{at } \theta_2\end{bmatrix}\theta_2(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ \theta_2 \text{ and } \theta_3\end{bmatrix}\theta_3(s) = \begin{bmatrix}\text{Sum of}\\ \text{applied torques}\\ \text{at } \theta_2\end{bmatrix} \tag{2.130b}$$

$$-\begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ \theta_1 \text{ and } \theta_3\end{bmatrix}\theta_1(s) - \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{between}\\ \theta_2 \text{ and } \theta_3\end{bmatrix}\theta_2(s) + \begin{bmatrix}\text{Sum of}\\ \text{impedances}\\ \text{connected}\\ \text{to the motion}\\ \text{at } \theta_3\end{bmatrix}\theta_3(s) = \begin{bmatrix}\text{Sum of}\\ \text{applied torques}\\ \text{at } \theta_3\end{bmatrix} \tag{2.130c}$$

Hence,

$$\begin{aligned}(J_1s^2 + D_1s + K)\theta_1(s) \quad &-K\theta_2(s) \quad &-0\theta_3(s) &= T(s)\\ -K\theta_1(s) + (J_2s^2 + D_2s + K)\theta_2(s) \quad &-D_2s\theta_3(s) &= 0\\ -0\theta_1(s) \quad &-D_2s\theta_2(s) + (J_3s^2 + D_3s + D_2s)\theta_3(s) &= 0\end{aligned} \tag{2.131a, b, c}$$

Skill-Assessment Exercise 2.9

PROBLEM: Find the transfer function, $G(s) = \theta_2(s)/T(s)$, for the rotational mechanical system shown in Figure 2.26.

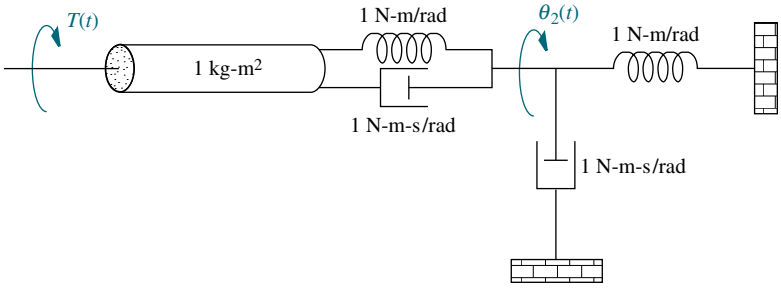


FIGURE 2.26 Rotational mechanical system for Skill-Assessment Exercise 2.9

ANSWER: $G(s) = \dfrac{1}{2s^2 + s + 1}$

The complete solution is at www.wiley.com/college/nise.

2.7 Transfer Functions for Systems with Gears

Now that we are able to find the transfer function for rotational systems, we realize that these systems, especially those driven by motors, are rarely seen without associated gear trains driving the load. This section covers this important topic.

Gears provide mechanical advantage to rotational systems. Anyone who has ridden a 10-speed bicycle knows the effect of gearing. Going uphill, you shift to provide more torque and less speed. On the straightaway, you shift to obtain more speed and less torque. Thus, gears allow you to match the drive system and the load—a trade-off between speed and torque.

For many applications, gears exhibit *backlash*, which occurs because of the loose fit between two meshed gears. The drive gear rotates through a small angle before making contact with the meshed gear. The result is that the angular rotation of the output gear does not occur until a small angular rotation of the input gear has occurred. In this section, we idealize the behavior of gears and assume that there is no backlash.

The linearized interaction between two gears is depicted in Figure 2.27. An input gear with radius r_1 and N_1 teeth is rotated through angle $\theta_1(t)$ due to a torque, $T_1(t)$. An output gear with radius r_2 and N_2 teeth responds by rotating through angle $\theta_2(t)$ and delivering a torque, $T_2(t)$. Let us now find the relationship between the rotation of Gear 1, $\theta_1(t)$, and Gear 2, $\theta_2(t)$.

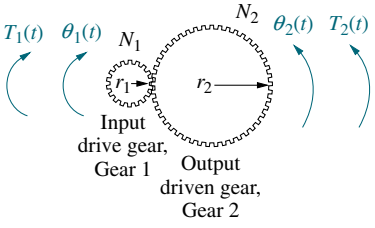


FIGURE 2.27 A gear system

From Figure 2.27, as the gears turn, the distance traveled along each gear's circumference is the same. Thus,

$$r_1\theta_1 = r_2\theta_2 \tag{2.132}$$

or

$$\boxed{\frac{\theta_2}{\theta_1} = \frac{r_1}{r_2} = \frac{N_1}{N_2}} \tag{2.133}$$

since the ratio of the number of teeth along the circumference is in the same proportion as the ratio of the radii. We conclude that the ratio of the angular displacement of the gears is inversely proportional to the ratio of the number of teeth.

What is the relationship between the input torque, T_1, and the delivered torque, T_2? If we assume the gears are *lossless*, that is they do not absorb or store energy, the energy into Gear 1 equals the energy out of Gear 2.[11] Since the translational energy of force times displacement becomes the rotational energy of torque times angular displacement,

$$\boxed{T_1\theta_1 = T_2\theta_2} \tag{2.134}$$

Solving Eq. (2.134) for the ratio of the torques and using Eq. (2.133), we get

$$\boxed{\frac{T_2}{T_1} = \frac{\theta_1}{\theta_2} = \frac{N_2}{N_1}} \tag{2.135}$$

Thus, the torques are directly proportional to the ratio of the number of teeth. All results are summarized in Figure 2.28.

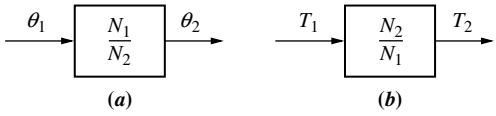


FIGURE 2.28 Transfer functions for **a.** angular displacement in lossless gears and **b.** torque in lossless gears

Let us see what happens to mechanical impedances that are driven by gears. Figure 2.29(*a*) shows gears driving a rotational inertia, spring, and viscous damper. For clarity, the gears are shown by an end-on view. We want to represent Figure 2.29(*a*) as an equivalent system at θ_1 without the gears. In other words, can the mechanical impedances be reflected from the output to the input, thereby eliminating the gears?

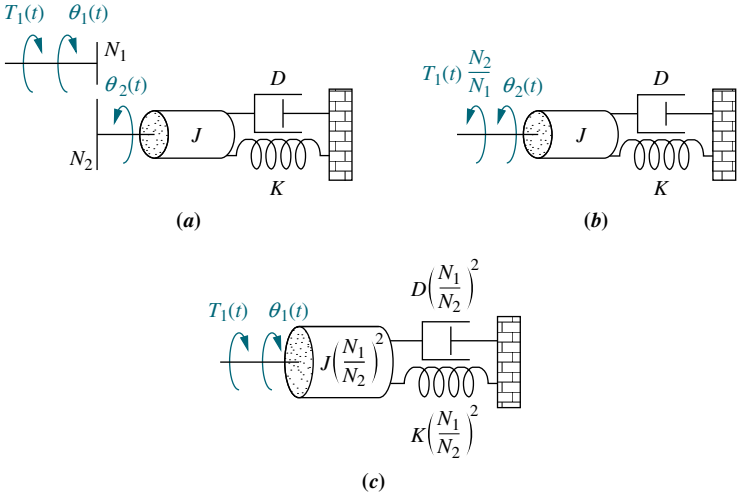


FIGURE 2.29 **a.** Rotational system driven by gears; **b.** equivalent system at the output after reflection of input torque; **c.** equivalent system at the input after reflection of impedances

[11] This is equivalent to saying that the gears have negligible inertia and damping.

From Figure 2.28(b), T_1 can be reflected to the output by multiplying by N_2/N_1. The result is shown in Figure 2.29(b), from which we write the equation of motion as

$$(Js^2 + Ds + K)\theta_2(s) = T_1(s)\frac{N_2}{N_1} \tag{2.136}$$

Now convert $\theta_2(s)$ into an equivalent $\theta_1(s)$, so that Eq. (2.136) will look as if it were written at the input. Using Figure 2.28(a) to obtain $\theta_2(s)$ in terms of $\theta_1(s)$, we get

$$(Js^2 + Ds + K)\frac{N_1}{N_2}\theta_1(s) = T_1(s)\frac{N_2}{N_1} \tag{2.137}$$

After simplification,

$$\left[J\left(\frac{N_1}{N_2}\right)^2 s^2 + D\left(\frac{N_1}{N_2}\right)^2 s + K\left(\frac{N_1}{N_2}\right)^2\right]\theta_1(s) = T_1(s) \tag{2.138}$$

which suggests the equivalent system at the input and without gears shown in Figure 2.29(c). Thus, the load can be thought of as having been reflected from the output to the input.

Generalizing the results, we can make the following statement: *Rotational mechanical impedances can be reflected through gear trains by multiplying the mechanical impedance by the ratio*

$$\left(\frac{\text{Number of teeth of gear on } \textit{destination} \text{ shaft}}{\text{Number of teeth of gear on } \textit{source} \text{ shaft}}\right)^2$$

where the impedance to be reflected is attached to the source shaft and is being reflected to the destination shaft. The next example demonstrates the application of the concept of reflected impedances as we find the transfer function of a rotational mechanical system with gears.

Example 2.21

Transfer Function—System with Lossless Gears

PROBLEM: Find the transfer function, $\theta_2(s)/T_1(s)$, for the system of Figure 2.30(a).

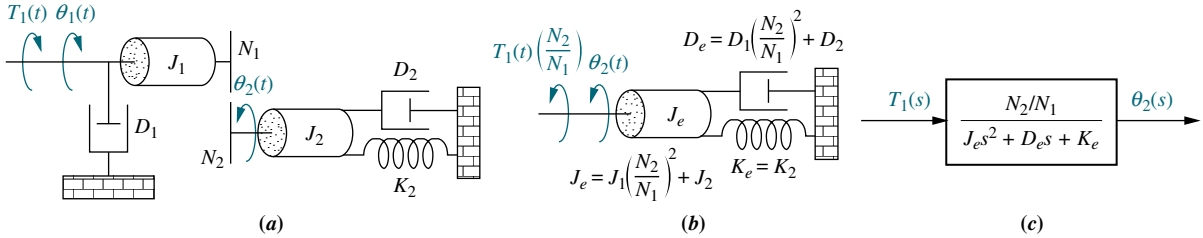


FIGURE 2.30 **a.** Rotational mechanical system with gears; **b.** system after reflection of torques and impedances to the output shaft; **c.** block diagram

SOLUTION: It may be tempting at this point to search for two simultaneous equations corresponding to each inertia. The inertias, however, do not undergo linearly independent motion, since they are tied together by the gears. Thus, there is only one degree of freedom and hence one equation of motion.

Let us first reflect the impedances (J_1 and D_1) and torque (T_1) on the input shaft to the output as shown in Figure 2.30(b), where the impedances are reflected by $(N_2/N_1)^2$ and the torque is reflected by (N_2/N_1). The equation of motion can now be written as

$$(J_e s^2 + D_e s + K_e)\theta_2(s) = T_1(s)\frac{N_2}{N_1} \tag{2.139}$$

where

$$J_e = J_1\left(\frac{N_2}{N_1}\right)^2 + J_2; \quad D_e = D_1\left(\frac{N_2}{N_1}\right)^2 + D_2; \quad K_e = K_2$$

Solving for $\theta_2(s)/T_1(s)$, the transfer function is found to be

$$G(s) = \frac{\theta_2(s)}{T_1(s)} = \frac{N_2/N_1}{J_e s^2 + D_e s + K_e} \tag{2.140}$$

as shown in Figure 2.30(c).

In order to eliminate gears with large radii, a *gear train* is used to implement large gear ratios by cascading smaller gear ratios. A schematic diagram of a gear train is shown in Figure 2.31. Next to each rotation, the angular displacement relative to θ_1 has been calculated. From Figure 2.31,

$$\theta_4 = \frac{N_1 N_3 N_5}{N_2 N_4 N_6}\theta_1 \tag{2.141}$$

For gear trains, we conclude that the equivalent gear ratio is the product of the individual gear ratios. We now apply this result to solve for the transfer function of a system that does not have lossless gears.

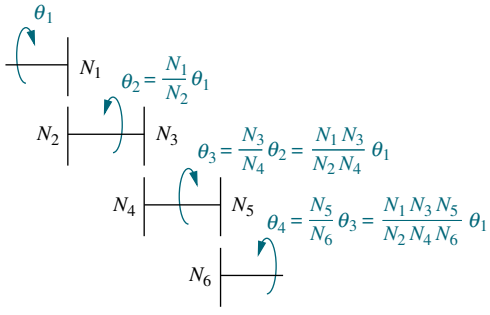


FIGURE 2.31 Gear train

Example 2.22

Transfer Function—Gears with Loss

PROBLEM: Find the transfer function, $\theta_1(s)/T_1(s)$, for the system of Figure 2.32(a).

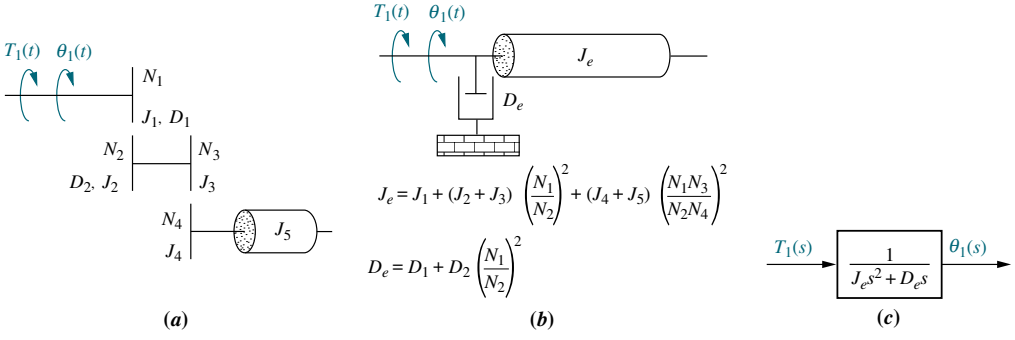


FIGURE 2.32
a. System using a gear train; **b.** equivalent system at the input; **c.** block diagram

SOLUTION: This system, which uses a gear train, does not have lossless gears. All of the gears have inertia, and for some shafts there is viscous friction. To solve the problem, we want to reflect all of the impedances to the input shaft, θ_1. The gear ratio is not the same for all impedances. For example, D_2 is reflected only through one gear ratio as $D_2(N_1/N_2)^2$, whereas J_4 plus J_5 is reflected through two gear ratios as $(J_4 + J_5)[(N_3/N_4)(N_1/N_2)]^2$. The result of reflecting all impedances to θ_1 is shown in Figure 2.32(b), from which the equation of motion is

$$(J_e s^2 + D_e s)\theta_1(s) = T_1(s) \tag{2.142}$$

where

$$J_e = J_1 + (J_2 + J_3)\left(\frac{N_1}{N_2}\right)^2 + (J_4 + J_5)\left(\frac{N_1 N_3}{N_2 N_4}\right)^2$$

and

$$D_e = D_1 + D_2\left(\frac{N_1}{N_2}\right)^2$$

From Eq. (2.142), the transfer function is

$$G(s) = \frac{\theta_1(s)}{T_1(s)} = \frac{1}{J_e s^2 + D_e s} \tag{2.143}$$

as shown in Figure 2.32(c).

Skill-Assessment Exercise 2.10

PROBLEM: Find the transfer function, $G(s) = \theta_2(s)/T(s)$, for the rotational mechanical system with gears shown in Figure 2.33.

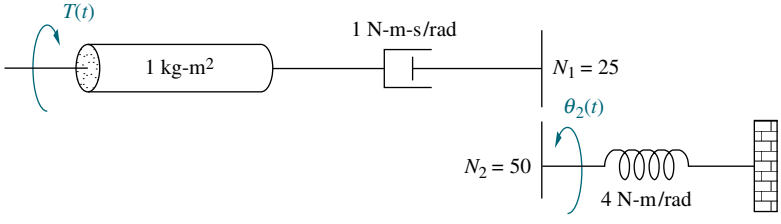


FIGURE 2.33 Rotational mechanical system with gears for Skill-Assessment Exercise 2.10

ANSWER: $G(s) = \dfrac{1/2}{s^2 + s + 1}$

The complete solution is at www.wiley.com/college/nise.

2.8 Electromechanical System Transfer Functions

In the last section we talked about rotational systems with gears, which completed our discussion of purely mechanical systems. Now, we move to systems that are hybrids of electrical and mechanical variables, the *electromechanical systems*. We have seen one application of an electromechanical system in Chapter 1, the antenna azimuth position control system. Other applications for systems with electromechanical components are robot controls, sun and star trackers, and computer tape and disk-drive position controls. An example of a control system that uses electromechanical components is shown in Figure 2.34.

A motor is an electromechanical component that yields a displacement output for a voltage input, that is, a mechanical output generated by an electrical input. We will derive the transfer function for one particular kind of electromechanical system, the armature-controlled dc servomotor (*Mablekos, 1980*). The motor's schematic is shown in Figure 2.35(*a*), and the transfer function we will derive appears in Figure 2.35(*b*).

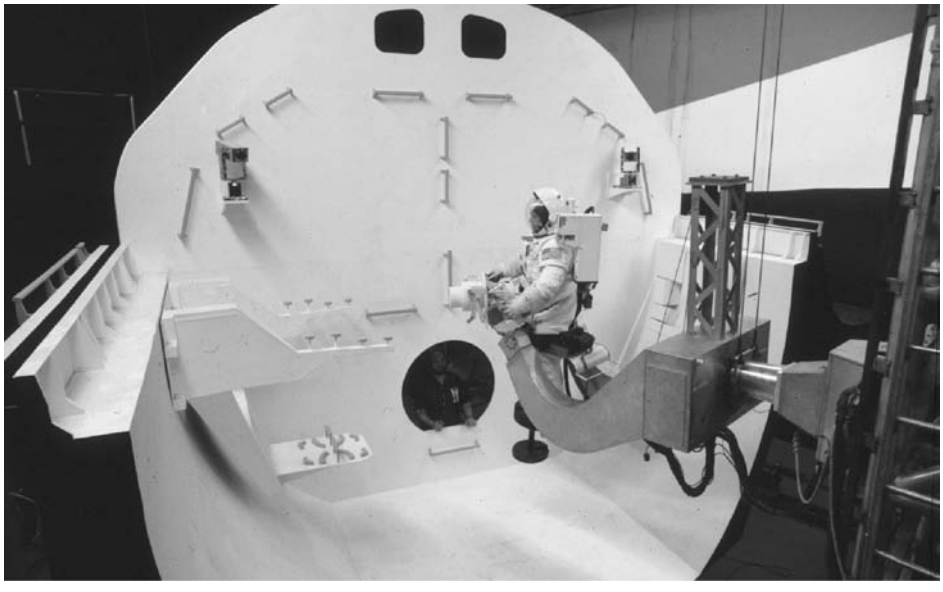


FIGURE 2.34 NASA flight simulator robot arm with electromechanical control system components.

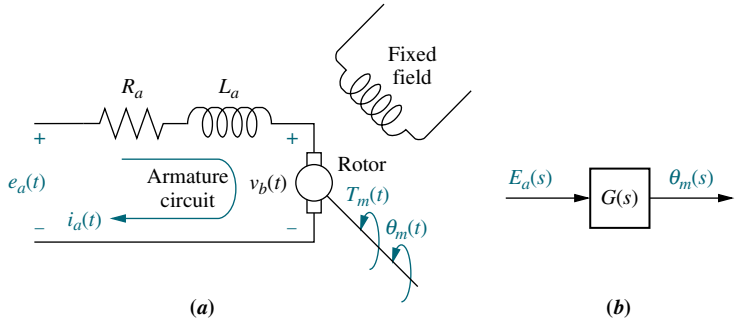


FIGURE 2.35 DC motor: **a.** schematic;[12] **b.** block diagram

[12] See Appendix I at www.wiley.com/college/nise for a derivation of this schematic and its parameters.

In Figure 2.35(a) a magnetic field is developed by stationary permanent magnets or a stationary electromagnet called the *fixed field.* A rotating circuit called the *armature*, through which current $i_a(t)$ flows, passes through this magnetic field at right angles and feels a force, $F = Bli_a(t)$, where B is the magnetic field strength and l is the length of the conductor. The resulting torque turns the *rotor*, the rotating member of the motor.

There is another phenomenon that occurs in the motor: A conductor moving at right angles to a magnetic field generates a voltage at the terminals of the conductor equal to $e = Blv$, where e is the voltage and v is the velocity of the conductor normal to the magnetic field. Since the current-carrying armature is rotating in a magnetic field, its voltage is proportional to speed. Thus,

$$v_b(t) = K_b \frac{d\theta_m(t)}{dt} \tag{2.144}$$

We call $v_b(t)$ the *back electromotive force (back emf);* K_b is a constant of proportionality called the back emf constant; and $d\theta_m(t)/dt = \omega_m(t)$ is the angular velocity of the motor. Taking the Laplace transform, we get

$$V_b(s) = K_b s\theta_m(s) \tag{2.145}$$

The relationship between the armature current, $i_a(t)$, the applied armature voltage, $e_a(t)$, and the back emf, $v_b(t)$, is found by writing a loop equation around the Laplace transformed armature circuit (see Figure 3.5(a)):

$$R_a I_a(s) + L_a s I_a(s) + V_b(s) = E_a(s) \tag{2.146}$$

The torque developed by the motor is proportional to the armature current; thus,

$$T_m(s) = K_t I_a(s) \tag{2.147}$$

where T_m is the torque developed by the motor, and K_t is a constant of proportionality, called the motor torque constant, which depends on the motor and magnetic field characteristics. In a consistent set of units, the value of K_t is equal to the value of K_b. Rearranging Eq. (2.147) yields

$$I_a(s) = \frac{1}{K_t} T_m(s) \tag{2.148}$$

To find the transfer function of the motor, we first substitute Eqs. (2.145) and (2.148) into (2.146), yielding

$$\frac{(R_a + L_a s)T_m(s)}{K_t} + K_b s\theta_m(s) = E_a(s) \tag{2.149}$$

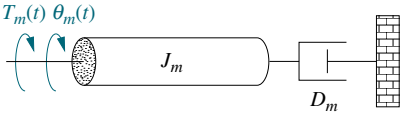


FIGURE 2.36 Typical equivalent mechanical loading on a motor

Now we must find $T_m(s)$ in terms of $\theta_m(s)$ if we are to separate the input and output variables and obtain the transfer function, $\theta_m(s)/E_a(s)$.

Figure 2.36 shows a typical equivalent mechanical loading on a motor. J_m is the equivalent inertia at the armature and includes both the armature inertia and, as we will see later, the load inertia reflected to the armature.

D_m is the equivalent viscous damping at the armature and includes both the armature viscous damping and, as we will see later, the load viscous damping reflected to the armature. From Figure 2.36,

$$T_m(s) = (J_m s^2 + D_m s)\theta_m(s) \tag{2.150}$$

Substituting Eq. (2.150) into Eq. (2.149) yields

$$\frac{(R_a + L_a s)(J_m s^2 + D_m s)\theta_m(s)}{K_t} + K_b s\theta_m(s) = E_a(s) \tag{2.151}$$

If we assume that the armature inductance, L_a, is small compared to the armature resistance, R_a, which is usual for a dc motor, Eq. (2.151) becomes

$$\left[\frac{R_a}{K_t}(J_m s + D_m) + K_b\right] s\theta_m(s) = E_a(s) \tag{2.152}$$

After simplification, the desired transfer function, $\theta_m(s)/E_a(s)$, is found to be

$$\boxed{\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/(R_a J_m)}{s\left[s + \frac{1}{J_m}(D_m + \frac{K_t K_b}{R_a})\right]}} \tag{2.153}$$[13]

Even though the form of Eq. (2.153) is relatively simple, namely

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K}{s(s+\alpha)} \tag{2.154}$$

the reader may be concerned about how to evaluate the constants.

Let us first discuss the mechanical constants, J_m and D_m. Consider Figure 2.37, which shows a motor with inertia J_a and damping D_a at the armature driving a load consisting of inertia J_L and damping D_L. Assuming that all inertia and damping values shown are known, J_L and D_L can be reflected back to the armature as some equivalent inertia and damping to be added to J_a and D_a, respectively. Thus, the equivalent inertia, J_m, and equivalent damping, D_m, at the armature are

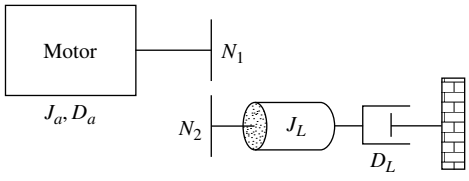


FIGURE 2.37 DC motor driving a rotational mechanical load

$$J_m = J_a + J_L\left(\frac{N_1}{N_2}\right)^2; \quad D_m = D_a + D_L\left(\frac{N_1}{N_2}\right)^2 \tag{2.155}$$[14]

Now that we have evaluated the mechanical constants, J_m and D_m, what about the electrical constants in the transfer function of Eq. (2.153)? We will show that these constants can be obtained through a *dynamometer* test of the motor, where a dynamometer measures the torque and speed of a motor under the condition of a

[13] The units for the electrical constants are K_t = N-m-A (newton-meters/ampere), and K_b = V-s/rad (volt-seconds/radian).

[14] If the values of the mechanical constants are not known, motor constants can be determined through laboratory testing using transient response or frequency response data. The concept of transient response is covered in Chapter 4; frequency response is covered in Chapter 10.

constant applied voltage. Let us first develop the relationships that dictate the use of a dynamometer.

Substituting Eqs. (2.145) and (2.148) into Eq. (2.146), with $L_a = 0$, yields

$$\frac{R_a}{K_t} T_m(s) + K_b s\theta_m(s) = E_a(s) \tag{2.156}$$

Taking the inverse Laplace transform, we get

$$\frac{R_a}{K_t} T_m(t) + K_b \omega_m(t) = e_a(t) \tag{2.157}$$

where the inverse Laplace transform of $s\theta_m(s)$ is $d\theta_m(t)/dt$ or, alternately, $\omega_m(t)$.

If a dc voltage, e_a, is applied, the motor will turn at a constant angular velocity, ω_m, with a constant torque, T_m. Hence, dropping the functional relationship based on time from Eq. (2.157), the following relationship exists when the motor is operating at steady state with a dc voltage input:

$$\frac{R_a}{K_t} T_m + K_b \omega_m = e_a \tag{2.158}$$

Solving for T_m yields

$$\boxed{T_m = -\frac{K_b K_t}{R_a} \omega_m + \frac{K_t}{R_a} e_a} \tag{2.159}$$

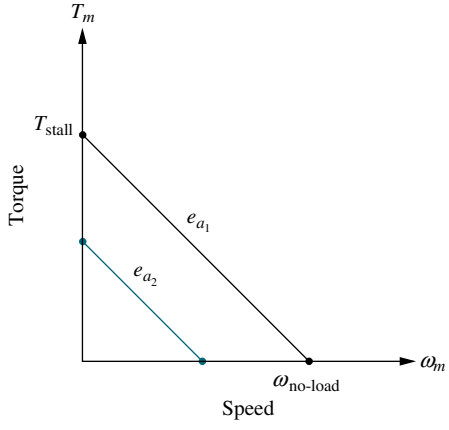


FIGURE 2.38 Torque-speed curves with an armature voltage, e_a, as a parameter

Equation (2.159) is a straight line, T_m vs. ω_m, and is shown in Figure 2.38. This plot is called the *torque-speed curve.* The torque axis intercept occurs when the angular velocity reaches zero. That value of torque is called the *stall torque*, T_{stall}. Thus,

$$T_{\text{stall}} = \frac{K_t}{R_a} e_a \tag{2.160}$$

The angular velocity occurring when the torque is zero is called the *no-load speed,* $\omega_{\text{no-load}}$. Thus,

$$\omega_{\text{no-load}} = \frac{e_a}{K_b} \tag{2.161}$$

The electrical constants of the motor's transfer function can now be found from Eqs. (2.160) and (2.161) as

$$\boxed{\frac{K_t}{R_a} = \frac{T_{\text{stall}}}{e_a}} \tag{2.162}$$

and

$$\boxed{K_b = \frac{e_a}{\omega_{\text{no-load}}}} \tag{2.163}$$

The electrical constants, K_t/R_a and K_b, can be found from a dynamometer test of the motor, which would yield T_{stall} and $\omega_{\text{no-load}}$ for a given e_a.

Example 2.23

Transfer Function—DC Motor and Load

PROBLEM: Given the system and torque-speed curve of Figure 2.39(*a*) and (*b*), find the transfer function, $\theta_L(s)/E_a(s)$.

SOLUTION: Begin by finding the mechanical constants, J_m and D_m, in Eq. (2.153). From Eq. (2.155), the total inertia at the armature of the motor is

$$J_m = J_a + J_L\left(\frac{N_1}{N_2}\right)^2 = 5 + 700\left(\frac{1}{10}\right)^2 = 12 \tag{2.164}$$

and the total damping at the armature of the motor is

$$D_m = D_a + D_L\left(\frac{N_1}{N_2}\right)^2 = 2 + 800\left(\frac{1}{10}\right)^2 = 10 \tag{2.165}$$

Now we will find the electrical constants, K_t/R_a and K_b. From the torque-speed curve of Figure 2.39(*b*),

$$T_{\text{stall}} = 500 \tag{2.166}$$

$$\omega_{\text{no-load}} = 50 \tag{2.167}$$

$$e_a = 100 \tag{2.168}$$

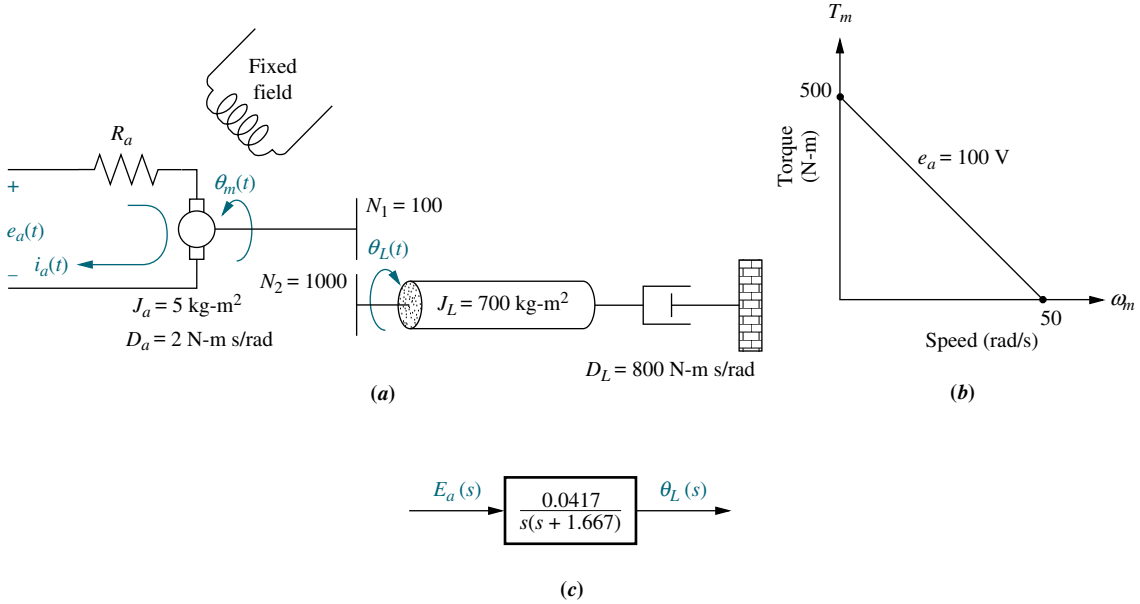


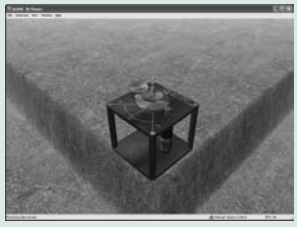
FIGURE 2.39 **a.** DC motor and load; **b.** torque-speed curve; **c.** block diagram

Hence the electrical constants are

$$\frac{K_t}{R_a} = \frac{T_{\text{stall}}}{e_a} = \frac{500}{100} = 5 \tag{2.169}$$

Virtual Experiment 2.2
Open-Loop Servo Motor

Put theory into practice exploring the dynamics of the Quanser Rotary Servo System modeled in LabVIEW. It is particularly important to know how a servo motor behaves when using them in high-precision applications such as hard disk drives.



Virtual experiments are found on WileyPLUS.

and

$$K_b = \frac{e_a}{\omega_{\text{no-load}}} = \frac{100}{50} = 2 \tag{2.170}$$

Substituting Eqs. (2.164), (2.165), (2.169), and (2.170) into Eq. (2.153) yield

$$\frac{\theta_m(s)}{E_a(s)} = \frac{5/12}{s\left\{s + \frac{1}{12}[10 + (5)(2)]\right\}} = \frac{0.417}{s(s + 1.667)} \tag{2.171}$$

In order to find $\theta_L(s)/E_a(s)$, we use the gear ratio, $N_1/N_2 = 1/10$, and find

$$\frac{\theta_L(s)}{E_a(s)} = \frac{0.0417}{s(s + 1.667)} \tag{2.172}$$

as shown in Figure 2.39(*c*).

Skill-Assessment Exercise 2.11

WileyPLUS
WPCS
Control Solutions

PROBLEM: Find the transfer function, $G(s) = \theta_L(s)/E_a(s)$, for the motor and load shown in Figure 2.40. The torque-speed curve is given by $T_m = -8\omega_m + 200$ when the input voltage is 100 volts.

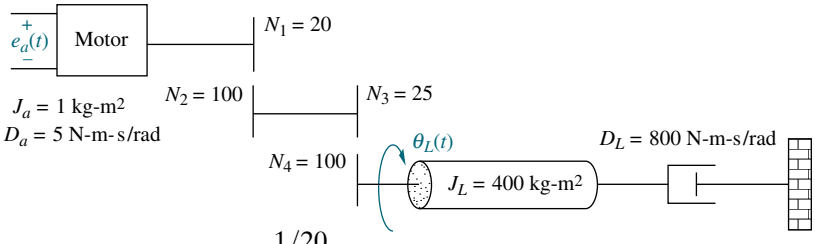


FIGURE 2.40 Electro-mechanical system for Skill-Assessment Exercise 2.11

ANSWER: $G(s) = \dfrac{1/20}{s[s + (15/2)]}$

The complete solution is at www.wiley.com/college/nise.

2.9 Electric Circuit Analogs

In this section, we show the commonality of systems from the various disciplines by demonstrating that the mechanical systems with which we worked can be represented by equivalent electric circuits. We have pointed out the similarity between the equations resulting from Kirchhoff's laws for electrical systems and the equations of motion of mechanical systems. We now show this commonality even more convincingly by producing electric circuit equivalents for mechanical systems. The variables of the electric circuits behave exactly as the analogous

variables of the mechanical systems. In fact, converting mechanical systems to electrical networks before writing the describing equations is a problem-solving approach that you may want to pursue.

An electric circuit that is analogous to a system from another discipline is called an electric circuit *analog*. Analogs can be obtained by comparing the describing equations, such as the equations of motion of a mechanical system, with either electrical mesh or nodal equations. When compared with mesh equations, the resulting electrical circuit is called a *series analog*. When compared with nodal equations, the resulting electrical circuit is called a *parallel analog*.

Series Analog

Consider the translational mechanical system shown in Figure 2.41(a), whose equation of motion is

$$(Ms^2 + f_v s + K)X(s) = F(s) \tag{2.173}$$

Kirchhoff's mesh equation for the simple series RLC network shown in Figure 2.41(b) is

$$\left(Ls + R + \frac{1}{Cs}\right)I(s) = E(s) \tag{2.174}$$

As we previously pointed out, Eq. (2.173) is not directly analogous to Eq. (2.174) because displacement and current are not analogous. We can create a direct analogy by operating on Eq. (2.173) to convert displacement to velocity by dividing and multiplying the left-hand side by s, yielding

$$\frac{Ms^2 + f_v s + K}{s} sX(s) = \left(Ms + f_v + \frac{K}{s}\right)V(s) = F(s) \tag{2.175}$$

Comparing Eqs. 2.174 and 2.175, we recognize the sum of impedances and draw the circuit shown in Figure 2.41(c). The conversions are summarized in Figure 2.41(d).

When we have more than one degree of freedom, the impedances associated with a motion appear as series electrical elements in a mesh, but

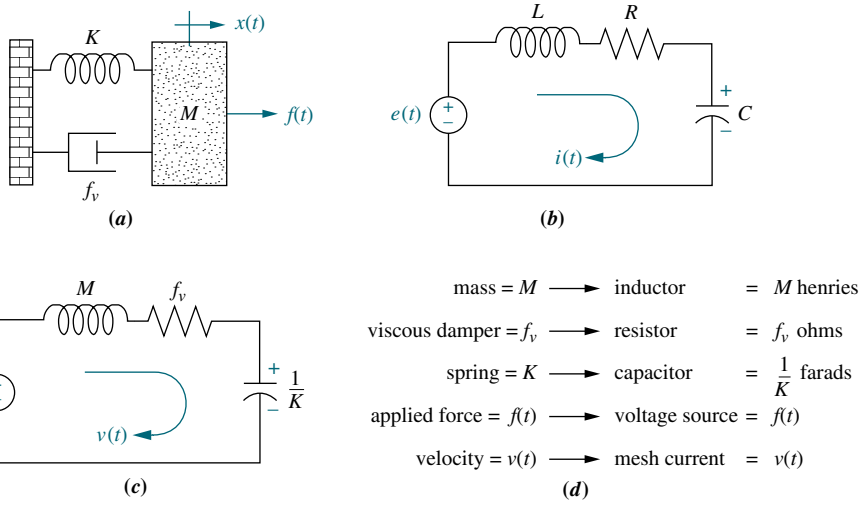


FIGURE 2.41 Development of series analog: **a.** mechanical system; **b.** desired electrical representation; **c.** series analog; **d.** parameters for series analog

the impedances between adjacent motions are drawn as series electrical impedances between the two corresponding meshes. We demonstrate with an example.

Example 2.24

Converting a Mechanical System to a Series Analog

PROBLEM: Draw a series analog for the mechanical system of Figure 2.17(a).

SOLUTION: Equations (2.118) are analogous to electrical mesh equations after conversion to velocity. Thus,

$$\left[M_1 s + (f_{v_1} + f_{v_3}) + \frac{(K_1 + K_2)}{s}\right] V_1(s) - \left(f_{v_3} + \frac{K_2}{s}\right) V_2(s) = F(s) \tag{2.176a}$$

$$-\left(f_{v_3} + \frac{K_2}{s}\right) V_1(s) + \left[M_2 s + (f_{v_2} + f_{v_3}) + \frac{(K_2 + K_3)}{s}\right] V_2(s) = 0 \tag{2.176b}$$

Coefficients represent sums of electrical impedance. Mechanical impedances associated with M_1 form the first mesh, where impedances between the two masses are common to the two loops. Impedances associated with M_2 form the second mesh. The result is shown in Figure 2.42, where $v_1(t)$ and $v_2(t)$ are the velocities of M_1 and M_2, respectively.

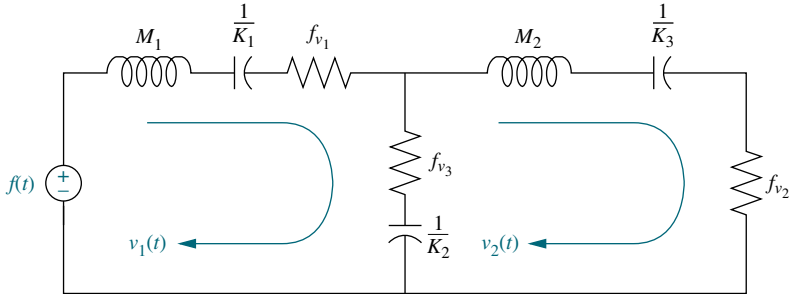


FIGURE 2.42 Series analog of mechanical system of Figure 2.17(a)

Parallel Analog

A system can also be converted to an equivalent parallel analog. Consider the translational mechanical system shown in Figure 2.43(a), whose equation of motion is given by Eq. (2.175). Kirchhoff's nodal equation for the simple parallel RLC network shown in Figure 2.43(b) is

$$\left(Cs + \frac{1}{R} + \frac{1}{Ls}\right) E(s) = I(s) \tag{2.177}$$

Comparing Eqs. (2.175) and (2.177), we identify the sum of admittances and draw the circuit shown in Figure 2.43(c). The conversions are summarized in Figure 2.43(d).

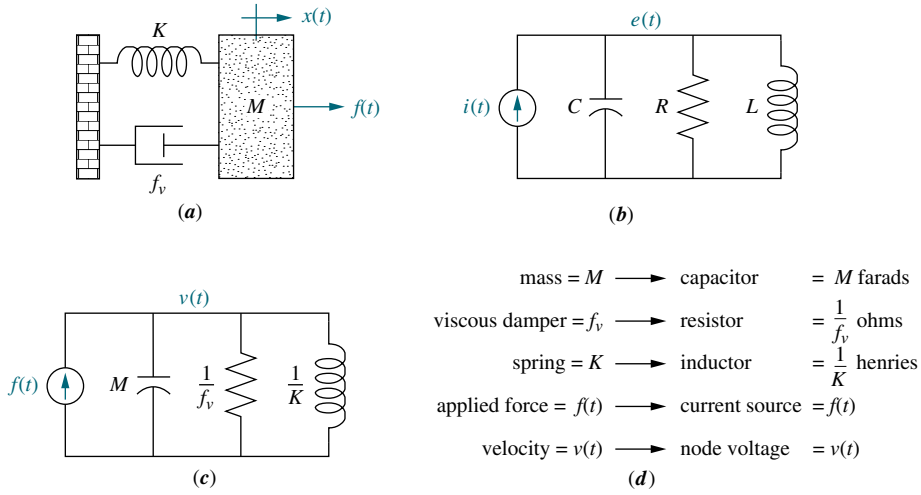


FIGURE 2.43 Development of parallel analog: **a.** mechanical system; **b.** desired electrical representation; **c.** parallel analog; **d.** parameters for parallel analog

When we have more than one degree of freedom, the components associated with a motion appear as parallel electrical elements connected to a node, but the components of adjacent motions are drawn as parallel electrical elements between two corresponding nodes. We demonstrate with an example.

Example 2.25

Converting a Mechanical System to a Parallel Analog

PROBLEM: Draw a parallel analog for the mechanical system of Figure 2.17(a).

SOLUTION: Equation (2.176) is also analogous to electrical node equations. Coefficients represent sums of electrical admittances. Admittances associated with M_1 form the elements connected to the first node, where mechanical admittances between the two masses are common to the two nodes. Mechanical admittances associated with M_2 form the elements connected to the second node. The result is shown in Figure 2.44, where $v_1(t)$ and $v_2(t)$ are the velocities of M_1 and M_2, respectively.

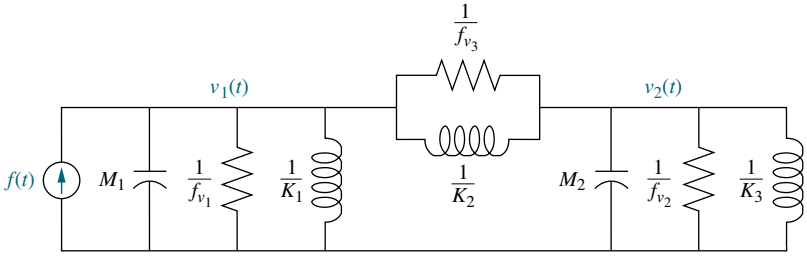


FIGURE 2.44 Parallel analog of mechanical system of Figure 2.17(a)

Skill-Assessment Exercise 2.12

PROBLEM: Draw a series and parallel analog for the rotational mechanical system of Figure 2.22.

ANSWER: The complete solution is at www.wiley.com/college/nise.

2.10 Nonlinearities

The models thus far are developed from systems that can be described approximately by linear, time-invariant differential equations. An assumption of *linearity* was implicit in the development of these models. In this section, we formally define the terms *linear* and *nonlinear* and show how to distinguish between the two. In Section 2.11, we show how to approximate a nonlinear system as a linear system so that we can use the modeling techniques previously covered in this chapter (*Hsu, 1968*).

A linear system possesses two properties: superposition and homogeneity. The property of *superposition* means that the output response of a system to the sum of inputs is the sum of the responses to the individual inputs. Thus, if an input of $r_1(t)$ yields an output of $c_1(t)$ and an input of $r_2(t)$ yields an output of $c_2(t)$, then an input of $r_1(t) + r_2(t)$ yields an output of $c_1(t) + c_2(t)$. The property of *homogeneity* describes the response of the system to a multiplication of the input by a scalar. Specifically, in a linear system, the property of homogeneity is demonstrated if for an input of $r_1(t)$ that yields an output of $c_1(t)$, an input of $Ar_1(t)$ yields an output of $Ac_1(t)$; that is, multiplication of an input by a scalar yields a response that is multiplied by the same scalar.

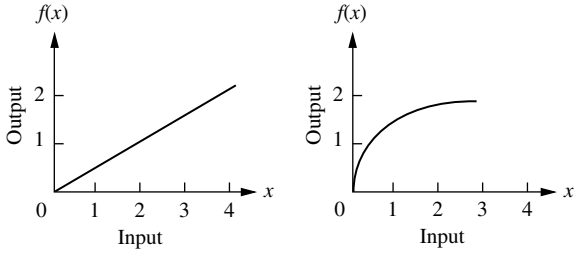


FIGURE 2.45 **a.** Linear system; **b.** nonlinear system

We can visualize linearity as shown in Figure 2.45. Figure 2.45(a) is a linear system where the output is always $\frac{1}{2}$ the input, or $f(x) = 0.5x$, regardless of the value of x. Thus each of the two properties of linear systems applies. For example, an input of 1 yields an output of $\frac{1}{2}$ and an input of 2 yields an output of 1. Using superposition, an input that is the sum of the original inputs, or 3, should yield an output that is the sum of the individual outputs, or 1.5. From Figure 2.45(a), an input of 3 does indeed yield an output of 1.5.

To test the property of homogeneity, assume an input of 2, which yields an output of 1. Multiplying this input by 2 should yield an output of twice as much, or 2. From Figure 2.45(a), an input of 4 does indeed yield an output of 2. The reader can verify that the properties of linearity certainly do not apply to the relationship shown in Figure 2.45(b).

Figure 2.46 shows some examples of physical nonlinearities. An electronic amplifier is linear over a specific range but exhibits the nonlinearity called *saturation* at high input voltages. A motor that does not respond at very low input voltages due to frictional forces exhibits a nonlinearity called *dead zone.* Gears that do not fit tightly exhibit a nonlinearity called *backlash:* The input moves over a small range

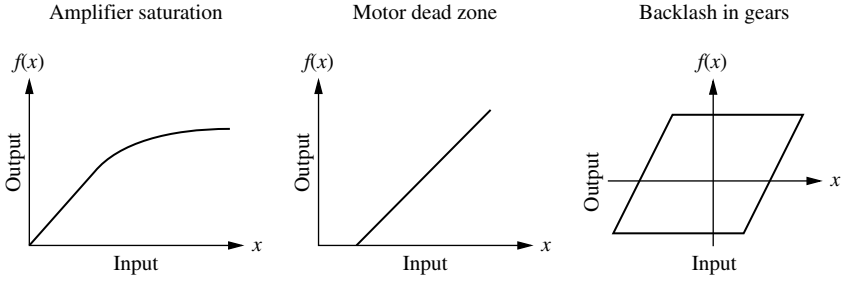


FIGURE 2.46 Some physical nonlinearities

without the output responding. The reader should verify that the curves shown in Figure 2.46 do not fit the definitions of linearity over their entire range. Another example of a nonlinear subsystem is a phase detector, used in a phase-locked loop in an FM radio receiver, whose output response is the sine of the input.

A designer can often make a linear approximation to a nonlinear system. Linear approximations simplify the analysis and design of a system and are used as long as the results yield a good approximation to reality. For example, a linear relationship can be established at a point on the nonlinear curve if the range of input values about that point is small and the origin is translated to that point. Electronic amplifiers are an example of physical devices that perform linear amplification with small excursions about a point.

2.11 Linearization

The electrical and mechanical systems covered thus far were assumed to be linear. However, if any nonlinear components are present, we must linearize the system before we can find the transfer function. In the last section, we defined and discussed nonlinearities; in this section, we show how to obtain linear approximations to nonlinear systems in order to obtain transfer functions.

The first step is to recognize the nonlinear component and write the nonlinear differential equation. When we linearize a nonlinear differential equation, we linearize it for small-signal inputs about the steady-state solution when the small-signal input is equal to zero. This steady-state solution is called *equilibrium* and is selected as the second step in the linearization process. For example, when a pendulum is at rest, it is at equilibrium. The angular displacement is described by a nonlinear differential equation, but it can be expressed with a linear differential equation for small excursions about this equilibrium point.

Next we linearize the nonlinear differential equation, and then we take the Laplace transform of the linearized differential equation, assuming zero initial conditions. Finally, we separate input and output variables and form the transfer function. Let us first see how to linearize a function; later, we will apply the method to the linearization of a differential equation.

If we assume a nonlinear system operating at point A, $[x_0, f(x_0)]$ in Figure 2.47, small changes in the input can be related to changes in the output about the point by way of the slope of the curve at the point A. Thus, if the slope of the curve at point A is m_a, then small excursions of the input about point A, δ_x, yield small changes in the output, $\delta f(x)$, related by the slope at point A. Thus,

$$[f(x) - f(x_0)] \approx m_a(x - x_0) \tag{2.178}$$

from which

$$\delta f(x) \approx m_a \delta x \tag{2.179}$$

and

$$f(x) \approx f(x_0) + m_a(x - x_0) \approx f(x_0) + m_a \delta x \tag{2.180}$$

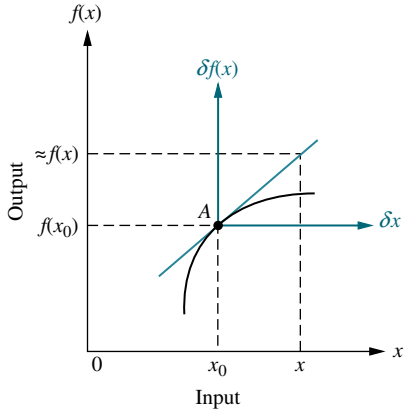


FIGURE 2.47 Linearization about point A

This relationship is shown graphically in Figure 2.47, where a new set of axes, δ_x and $\delta f(x)$, is created at the point A, and $f(x)$ is approximately equal to $f(x_0)$, the ordinate of the new origin, plus small excursions, $m_a \delta x$, away from point A. Let us look at an example.

Example 2.26

Linearizing a Function

PROBLEM: Linearize $f(x) = 5\cos x$ about $x = \pi/2$.

SOLUTION: We first find that the derivative of $f(x)$ is $df/dx = (-5\sin x)$. At $x = \pi/2$, the derivative is -5. Also $f(x_0) = f(\pi/2) = 5\cos(\pi/2) = 0$. Thus, from Eq. (2.180), the system can be represented as $f(x) = -5\,\delta x$ for small excursions of x about $\pi/2$. The process is shown graphically in Figure 2.48, where the cosine curve does indeed look like a straight line of slope -5 near $\pi/2$.

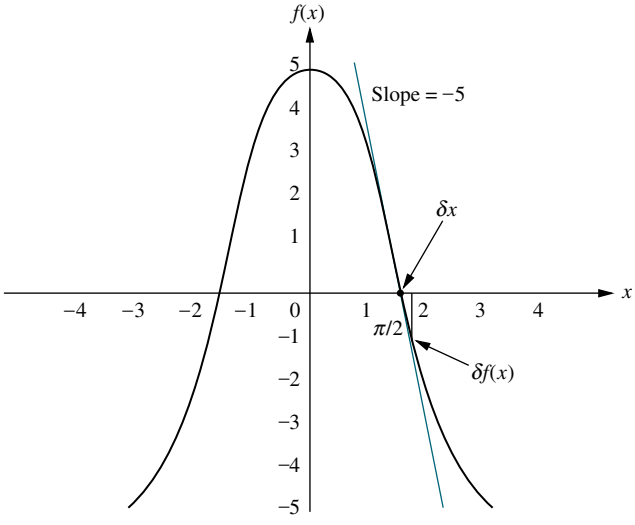


FIGURE 2.48 Linearization of $5\cos x$ about $x = \pi/2$

The previous discussion can be formalized using the Taylor series expansion, which expresses the value of a function in terms of the value of that function at a particular point, the excursion away from that point, and derivatives evaluated at that point. The Taylor series is shown in Eq. (2.181).

$$f(x) = f(x_0) + \left.\frac{df}{dx}\right|_{x=x_0} \frac{(x - x_0)}{1!} + \left.\frac{d^2 f}{dx^2}\right|_{x=x_0} \frac{(x - x_0)^2}{2!} + \cdots \tag{2.181}$$

For small excursions of x from x_0, we can neglect higher-order terms. The resulting approximation yields a straight-line relationship between the change in $f(x)$ and the excursions away from x_0. Neglecting the higher-order terms in Eq. (2.181), we get

$$f(x) - f(x_0) \approx \left.\frac{df}{dx}\right|_{x=x_0} (x - x_0) \tag{2.182}$$

or

$$\boxed{\delta f(x) \approx m|_{x=x_0}\delta x} \tag{2.183}$$

which is a linear relationship between $\delta f(x)$ and δx for small excursions away from x_0. It is interesting to note that Eqs. (2.182) and (2.183) are identical to Eqs. (2.178) and (2.179), which we derived intuitively. The following examples illustrate linearization. The first example demonstrates linearization of a differential equation, and the second example applies linearization to finding a transfer function.

Example 2.27

Linearizing a Differential Equation

PROBLEM: Linearize Eq. (2.184) for small excursions about $x = \pi/4$.

$$\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + \cos x = 0 \tag{2.184}$$

SOLUTION: The presence of the term $\cos x$ makes this equation nonlinear. Since we want to linearize the equation about $x = \pi/4$, we let $x = \delta x + \pi/4$, where δx is the small excursion about $\pi/4$, and substitute x into Eq. (2.184):

$$\frac{d^2\left(\delta x + \frac{\pi}{4}\right)}{dt^2} + 2\frac{d\left(\delta x + \frac{\pi}{4}\right)}{dt} + \cos\left(\delta x + \frac{\pi}{4}\right) = 0 \tag{2.185}$$

But

$$\frac{d^2\left(\delta x + \frac{\pi}{4}\right)}{dt^2} = \frac{d^2\delta x}{dt^2} \tag{2.186}$$

and

$$\frac{d\left(\delta x + \frac{\pi}{4}\right)}{dt} = \frac{d\delta x}{dt} \tag{2.187}$$

Finally, the term $\cos(\delta x + (\pi/4))$ can be linearized with the truncated Taylor series. Substituting $f(x) = \cos(\delta x + (\pi/4))$, $f(x_0) = f(\pi/4) = \cos(\pi/4)$, and $(x - x_0) = \delta x$ into Eq. (2.182) yields

$$\cos\left(\delta x + \frac{\pi}{4}\right) - \cos\left(\frac{\pi}{4}\right) = \left.\frac{d\cos x}{dx}\right|_{x=\frac{\pi}{4}} \delta x = -\sin\left(\frac{\pi}{4}\right)\delta x \tag{2.188}$$

Solving Eq. (2.188) for $\cos(\delta x + (\pi/4))$, we get

$$\cos\left(\delta x + \frac{\pi}{4}\right) = \cos\left(\frac{\pi}{4}\right) - \sin\left(\frac{\pi}{4}\right)\delta x = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\delta x \tag{2.189}$$

Substituting Eqs. (2.186), (2.187), and (2.189) into Eq. (2.185) yields the following linearized differential equation:

$$\frac{d^2\delta x}{dt^2} + 2\frac{d\delta x}{dt} - \frac{\sqrt{2}}{2}\delta x = -\frac{\sqrt{2}}{2} \tag{2.190}$$

This equation can now be solved for δx, from which we can obtain $x = \delta x + (\pi/4)$.

Even though the nonlinear Eq. (2.184) is homogeneous, the linearized Eq. (2.190) is not homogeneous. Eq. (2.190) has a forcing function on its right-hand side. This additional term can be thought of as an input to a system represented by Eq. (2.184).

Another observation about Eq. (2.190) is the negative sign on the left-hand side. The study of differential equations tells us that since the roots of the characteristic equation are positive, the homogeneous solution grows without bound instead of diminishing to zero. Thus, this system linearized around $x = \pi/4$ is not stable.

Example 2.28

Transfer Function—Nonlinear Electrical Network

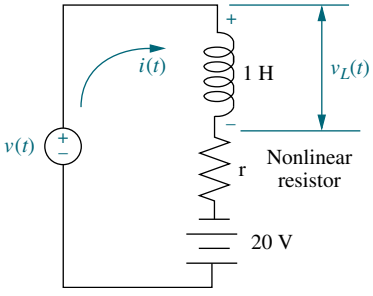


FIGURE 2.49 Nonlinear electrical network

PROBLEM: Find the transfer function, $V_L(s)/V(s)$, for the electrical network shown in Figure 2.49, which contains a nonlinear resistor whose voltage-current relationship is defined by $i_r = 2e^{0.1v_r}$, where i_r and v_r are the resistor current and voltage, respectively. Also, $v(t)$ in Figure 2.49 is a small-signal source.

SOLUTION: We will use Kirchhoff's voltage law to sum the voltages in the loop to obtain the nonlinear differential equation, but first we must solve for the voltage across the nonlinear resistor. Taking the natural log of the resistor's current-voltage relationship, we get $v_r = 10 \ln \frac{1}{2} i_r$. Applying Kirchhoff's voltage law around the loop, where $i_r = i$, yields

$$L\frac{di}{dt} + 10\ln\frac{1}{2}i - 20 = v(t) \tag{2.191}$$

Next, letus evaluate the equilibrium solution. First, set the small-signal source, $v(t)$, equal to zero. Now evaluate the steady-state current. With $v(t) = 0$, the circuit consists of a 20 V battery in series with the inductor and nonlinear resistor. In the steady state, the voltage across the inductor will be zero, since $v_L(t) = Ldi/dt$ and di/dt is zero in the steady state, given a constant battery source. Hence, the resistor voltage, v_r, is 20 V. Using the characteristics of the resistor, $i_r = 2e^{0.1v_r}$, we find that $i_r = i = 14.78$ amps. This current, i_0, is the equilibrium value of the network current. Hence $i = i_0 + \delta i$. Substituting this current into Eq. (2.191) yields

$$L\frac{d(i_0 + \delta i)}{dt} + 10\ln\frac{1}{2}(i_0 + \delta i) - 20 = v(t) \tag{2.192}$$

Using Eq. (2.182) to linearize $\ln\frac{1}{2}(i_0 + \delta i)$, we get

$$\ln\frac{1}{2}(i_0 + \delta i) - \ln\frac{1}{2}i_0 = \frac{d\left(\ln\frac{1}{2}i\right)}{di}\bigg|_{i=i_0} \delta i = \frac{1}{i}\bigg|_{i=i_0} \delta i = \frac{1}{i_0}\delta i \tag{2.193}$$

or

$$\ln\frac{1}{2}(i_0 + \delta i) = \ln\frac{i_0}{2} + \frac{1}{i_0}\delta i \tag{2.194}$$

Substituting into Eq. (2.192), the linearized equation becomes

$$L\frac{d\delta i}{dt} + 10\left(\ln\frac{i_0}{2} + \frac{1}{i_0}\delta i\right) - 20 = v(t) \tag{2.195}$$

Letting $L = 1$ and $i_0 = 14.78$, the final linearized differential equation is

$$\frac{d\delta i}{dt} + 0.677\delta i = v(t) \tag{2.196}$$

Taking the Laplace transform with zero initial conditions and solving for $\delta i(s)$, we get

$$\delta i(s) = \frac{V(s)}{s + 0.677} \tag{2.197}$$

But the voltage across the inductor about the equilibrium point is

$$v_L(t) = L\frac{d}{dt}(i_0 + \delta i) = L\frac{d\delta i}{dt} \tag{2.198}$$

Taking the Laplace transform,

$$V_L(s) = Ls\delta i(s) = s\delta i(s) \tag{2.199}$$

Substituting Eq. (2.197) into Eq. (2.199) yields

$$V_L(s) = s\frac{V(s)}{s + 0.677} \tag{2.200}$$

from which the final transfer function is

$$\frac{V_L(s)}{V(s)} = \frac{s}{s + 0.677} \tag{2.201}$$

for small excursions about $i = 14.78$ or, equivalently, about $v(t) = 0$.

Skill-Assessment Exercise 2.13

PROBLEM: Find the linearized transfer function, $G(s) = V(s)/I(s)$, for the electrical network shown in Figure 2.50. The network contains a nonlinear resistor whose voltage-current relationship is defined by $i_r = e^{v_r}$. The current source, $i(t)$, is a small-signal generator.

WileyPLUS
WPCS
Control Solutions

ANSWER: $G(s) = \dfrac{1}{s + 2}$

The complete solution is at www.wiley.com/college/nise.

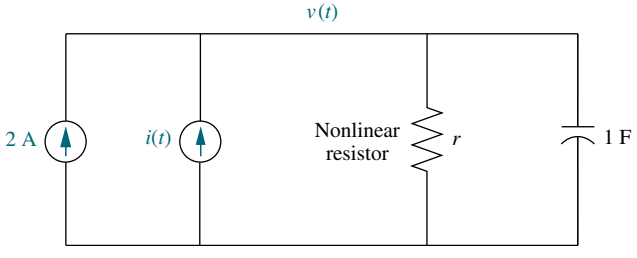


FIGURE 2.50 Nonlinear electric circuit for Skill-Assessment Exercise 2.13

Case Studies

Antenna Control: Transfer Functions

This chapter showed that physical systems can be modeled mathematically with transfer functions. Typically, systems are composed of subsystems of different types, such as electrical, mechanical, and electromechanical.

The first case study uses our ongoing example of the antenna azimuth position control system to show how to represent each subsystem as a transfer function.

PROBLEM: Find the transfer function for each subsystem of the antenna azimuth position control system schematic shown on the front endpapers. Use Configuration 1.

SOLUTION: First, we identify the individual subsystems for which we must find transfer functions; they are summarized in Table 2.6. We proceed to find the transfer function for each subsystem.

TABLE 2.6 Subsystems of the antenna azimuth position control system

Subsystem	Input	Output
Input potentiometer	Angular rotation from user, $\theta_i(t)$	Voltage to preamp, $v_i(t)$
Preamp	Voltage from potentiometers, $v_e(t) = v_i(t) - v_0(t)$	Voltage to power amp, $v_p(t)$
Power amp	Voltage from preamp, $v_p(t)$	Voltage to motor, $e_a(t)$
Motor	Voltage from power amp, $e_a(t)$	Angular rotation to load, $\theta_0(t)$
Output potentiometer	Angular rotation from load, $\theta_0(t)$	Voltage to preamp, $v_0(t)$

Input Potentiometer; Output Potentiometer

Since the input and output potentiometers are configured in the same way, their transfer functions will be the same. We *neglect* the dynamics for the potentiometers and simply find the relationship between the output voltage and the input angular displacement. In the center position the output voltage is zero. Five turns toward either the positive 10 volts or the negative 10 volts yields a voltage change of 10 volts. Thus, the transfer function, $V_i(s)/\theta_i(s)$, for the potentiometers is found by dividing the voltage change by the angular displacement:

$$\frac{V_i(s)}{\theta_i(s)} = \frac{10}{10\pi} = \frac{1}{\pi} \tag{2.202}$$

Preamplifier; Power Amplifier

The transfer functions of the amplifiers are given in the problem statement. Two phenomena are *neglected.* First, we *assume* that saturation is never reached. Second, the dynamics of the preamplifier are *neglected*, since its speed of response is typically much greater than that of the power amplifier. The transfer functions of both amplifiers are given in the problem statement and are the ratio of the Laplace transforms of the output voltage divided by the input voltage. Hence, for the

preamplifier,

$$\frac{V_p(s)}{V_e(s)} = K \tag{2.203}$$

and for the power amplifier,

$$\frac{E_a(s)}{V_p(s)} = \frac{100}{s+100} \tag{2.204}$$

Motor and Load

The motor and its load are next. The transfer function relating the armature displacement to the armature voltage is given in Eq. (2.153). The equivalent inertia, J_m, is

$$J_m = J_a + J_L\left(\frac{25}{250}\right)^2 = 0.02 + 1\frac{1}{100} = 0.03 \tag{2.205}$$

where $J_L = 1$ is the load inertia at θ_0. The equivalent viscous damping, D_m, at the armature is

$$D_m = D_a + D_L\left(\frac{25}{250}\right)^2 = 0.01 + 1\frac{1}{100} = 0.02 \tag{2.206}$$

where D_L is the load viscous damping at θ_0. From the problem statement, $K_t = 0.5$ N-m/A, $K_b = 0.5$ V-s/rad, and the armature resistance $R_a = 8$ ohms. These quantities along with J_m and D_m are substituted into Eq. (2.153), yielding the transfer function of the motor from the armature voltage to the armature displacement, or

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/(R_aJ_m)}{s\left[s + \frac{1}{J_m}\left(D_m + \frac{K_tK_b}{R_a}\right)\right]} = \frac{2.083}{s(s+1.71)} \tag{2.207}$$

To complete the transfer function of the motor, we multiply by the gear ratio to arrive at the transfer function relating load displacement to armature voltage:

$$\frac{\theta_0(s)}{E_a(s)} = 0.1\frac{\theta_m(s)}{E_a(s)} = \frac{0.2083}{s(s+1.71)} \tag{2.208}$$

The results are summarized in the block diagram and table of block diagram parameters (Configuration 1) shown on the front endpapers.

CHALLENGE: We now give you a problem to test your knowledge of this chapter's objectives: Referring to the antenna azimuth position control system schematic shown on the front endpapers, evaluate the transfer function of each subsystem. Use Configuration 2. Record your results in the table of block diagram parameters shown on the front endpapers for use in subsequent chapters' case study challenges.

Transfer Function of a Human Leg

In this case study we find the transfer function of a biological system. The system is a human leg, which pivots from the hip joint. In this problem, the component of weight is nonlinear, so the system requires linearization before the evaluation of the transfer function.

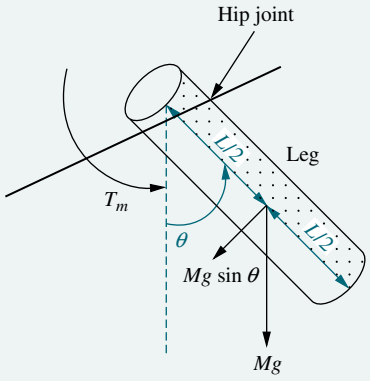


FIGURE 2.51 Cylinder model of a human leg.

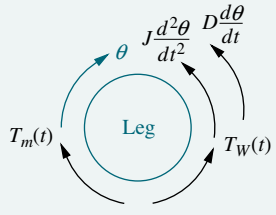


FIGURE 2.52 Free-body diagram of leg model

PROBLEM: The transfer function of a human leg relates the output angular rotation about the hip joint to the input torque supplied by the leg muscle. A simplified model for the leg is shown in Figure 2.51. The model *assumes* an applied muscular torque, $T_m(t)$, viscous damping, D, at the hip joint, and inertia, J, around the hip joint.[15] Also, a component of the weight of the leg, Mg, where M is the mass of the leg and g is the acceleration due to gravity, creates a nonlinear torque. If we *assume* that the leg is of uniform density, the weight can be applied at $L/2$, where L is the length of the leg (*Milsum, 1966*). Do the following:

a. Evaluate the nonlinear torque.

b. Find the transfer function, $\theta(s)/T_m(s)$, for small angles of rotation, where $\theta(s)$ is the angular rotation of the leg about the hip joint.

SOLUTION: First, calculate the torque due to the weight. The total weight of the leg is Mg acting vertically. The component of the weight in the direction of rotation is $Mg \sin\theta$. This force is applied at a distance $L/2$ from the hip joint. Hence the torque in the direction of rotation, $T_W(t)$, *is* $Mg(L/2)\sin\theta$. Next, draw a free-body diagram of the leg, showing the applied torque, $T_m(t)$, the torque due to the weight, $T_W(t)$, and the opposing torques due to inertia and viscous damping (see Figure 2.52).

Summing torques, we get

$$J\frac{d^2\theta}{dt^2} + D\frac{d\theta}{dt} + Mg\frac{L}{2}\sin\theta = T_m(t) \tag{2.209}$$

We linearize the system about the equilibrium point, $\theta = 0$, the vertical position of the leg. Using Eq. (2.182), we get

$$\sin\theta - \sin 0 = (\cos 0)\delta\theta \tag{2.210}$$

from which, $\sin\theta = \delta\theta$. Also, $J\,d^2\theta/dt^2 = J\,d^2\delta\theta/dt^2$ and $D\,d\theta/dt = D\,d\delta\theta/dt$. Hence Eq. (2.209) becomes

$$J\frac{d^2\delta\theta}{dt^2} + D\frac{d\delta\theta}{dt} + Mg\frac{L}{2}\delta\theta = T_m(t) \tag{2.211}$$

Notice that the torque due to the weight approximates a spring torque on the leg. Taking the Laplace transform with zero initial conditions yields

$$\left(Js^2 + Ds + Mg\frac{L}{2}\right)\delta\theta(s) = T_m(s) \tag{2.212}$$

from which the transfer function is

$$\frac{\delta\theta(s)}{T_m(s)} = \frac{1/J}{s^2 + \frac{D}{J}s + \frac{MgL}{2J}} \tag{2.213}$$

[15] For emphasis, J is not around the center of mass, as we previously assumed for inertia in mechanical rotation.

for small excursions about the equilibrium point, $\theta = 0$.

CHALLENGE: We now introduce a case study challenge to test your knowledge of this chapter's objectives. Although the physical system is different from a human leg, the problem demonstrates the same principles: linearization followed by transfer function evaluation.

Given the nonlinear electrical network shown in Figure 2.53, find the transfer function relating the output nonlinear resistor voltage, $V_r(s)$, to the input source voltage, $V(s)$.

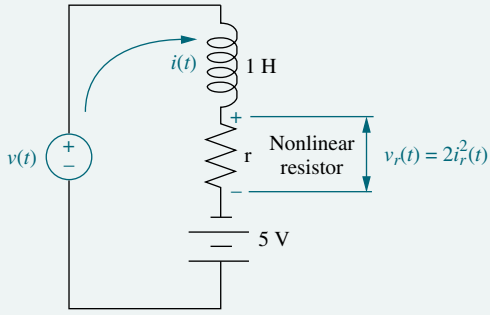


FIGURE 2.53 Nonlinear electric circuit

Summary

In this chapter, we discussed how to find a mathematical model, called a *transfer function*, for linear, time-invariant electrical, mechanical, and electromechanical systems. The transfer function is defined as $G(s) = C(s)/R(s)$, or the ratio of the Laplace transform of the output to the Laplace transform of the input. This relationship is algebraic and also adapts itself to modeling interconnected subsystems.

We realize that the physical world consists of more systems than we illustrated in this chapter. For example, we could apply transfer function modeling to hydraulic, pneumatic, heat, and even economic systems. Of course, we must assume these systems to be linear, or make linear approximations, in order to use this modeling technique.

Now that we have our transfer function, we can evaluate its response to a specified input. System response will be covered in Chapter 4. For those pursuing the state-space approach, we continue our discussion of modeling in Chapter 3, where we use the time domain rather than the frequency domain.

Review Questions

1. What mathematical model permits easy interconnection of physical systems?
2. To what classification of systems can the transfer function be best applied?
3. What transformation turns the solution of differential equations into algebraic manipulations?
4. Define the transfer function.
5. What assumption is made concerning initial conditions when dealing with transfer functions?
6. What do we call the mechanical equations written in order to evaluate the transfer function?
7. If we understand the form the mechanical equations take, what step do we avoid in evaluating the transfer function?
8. Why do transfer functions for mechanical networks look identical to transfer functions for electrical networks?

9. What function do gears perform?
10. What are the component parts of the mechanical constants of a motor's transfer function?
11. The motor's transfer function relates armature displacement to armature voltage. How can the transfer function that relates load displacement and armature voltage be determined?
12. Summarize the steps taken to linearize a nonlinear system.

Problems

1. Derive the Laplace transform for the following time functions: [Section: 2.2]
   a. $u(t)$
   b. $tu(t)$
   c. $\sin \omega t\, u(t)$
   d. $\cos \omega t\, u(t)$

2. Using the Laplace transform pairs of Table 2.1 and the Laplace transform theorems of Table 2.2, derive the Laplace transforms for the following time functions: [Section: 2.2]
   a. $e^{-at}\sin \omega t\, u(t)$
   b. $e^{-at}\cos \omega t\, u(t)$
   c. $t^3 u(t)$

WileyPLUS WPCS Control Solutions

3. Repeat Problem 18 in Chapter 1, using Laplace transforms. Assume that the forcing functions are zero prior to $t = 0-$. [Section: 2.2]
4. Repeat Problem 19 in Chapter 1, using Laplace transforms. Use the following initial conditions for each part as follows: (a) $x(0) = 4$, $x'(0) = -4$; (b) $x(0) = 4$, $x'(0) = 1$; (c) $x(0) = 2$, $x'(0) = 3$, where $x'(0) = \dfrac{dx}{dt}(0)$. Assume that the forcing functions are zero prior to $t = 0-$. [Section: 2.2]
5. Use MATLAB and the Symbolic Math Toolbox to find the Laplace transform of the following time functions: [Section: 2.2]
   a. $f(t) = 8t^2\cos(3t + 45°)$
   b. $f(t) = 3te^{-2t}\sin(4t + 60°)$

Symbolic Math SM

6. Use MATLAB and the Symbolic Math Toolbox to find the inverse Laplace transform of the following frequency functions: [Section: 2.2]
   a. $G(s) = \dfrac{(s^2 + 3s + 10)(s + 5)}{(s + 3)(s + 4)(s^2 + 2s + 100)}$
   b. $G(s) = \dfrac{s^3 + 4s^2 + 2s + 6}{(s + 8)(s^2 + 8s + 3)(s^2 + 5s + 7)}$

Symbolic Math SM

7. A system is described by the following differential equation:

$$\frac{d^3y}{dt^3} + 3\frac{d^2y}{dt^2} + 5\frac{dy}{dt} + y = \frac{d^3x}{dt^3} + 4\frac{d^2x}{dt^2} + 6\frac{dx}{dt} + 8x$$

   Find the expression for the transfer function of the system, $Y(s)/X(s)$. [Section: 2.3]
8. For each of the following transfer functions, write the corresponding differential equation. [Section: 2.3]
   a. $\dfrac{X(s)}{F(s)} = \dfrac{7}{s^2 + 5s + 10}$
   b. $\dfrac{X(s)}{F(s)} = \dfrac{15}{(s + 10)(s + 11)}$
   c. $\dfrac{X(s)}{F(s)} = \dfrac{s + 3}{s^3 + 11s^2 + 12s + 18}$
9. Write the differential equation for the system shown in Figure P2.1. [Section: 2.3]

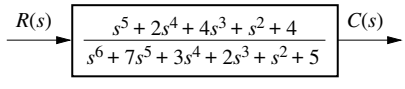


FIGURE P2.1

10. Write the differential equation that is mathematically equivalent to the

WileyPLUS WPCS Control Solutions

block diagram shown in Figure P2.2. Assume that $r(t) = 3t^3$. [Section: 2.3]

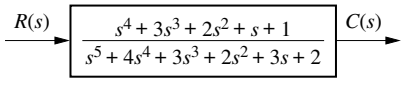


FIGURE P2.2

11. A system is described by the following differential equation: [Section 2.3]

$$\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + 3x = 1$$

with the initial conditions $x(0) = 1$, $\dot{x}(0) = -1$. Show a block diagram of the system, giving its transfer function and all pertinent inputs and outputs. (Hint: the initial conditions will show up as added inputs to an effective system with zero initial conditions.)

12. Use MATLAB to generate the transfer function: [Section: 2.3]

MATLAB ML

$$G(s) = \frac{5(s+15)(s+26)(s+72)}{s(s+55)(s^2+5s+30)(s+56)(s^2+27s+52)}$$

in the following ways:

a. the ratio of factors;

b. the ratio of polynomials.

13. Repeat Problem 12 for the following transfer function: [Section: 2.3]

MATLAB ML

$$G(s) = \frac{s^4 + 25s^3 + 20s^2 + 15s + 42}{s^5 + 13s^4 + 9s^3 + 37s^2 + 35s + 50}$$

14. Use MATLAB to generate the partial-fraction expansion of the following function: [Section: 2.3]

$$F(s) = \frac{10^4(s+5)(s+70)}{s(s+45)(s+55)(s^2+7s+110)(s^2+6s+95)}$$

15. Use MATLAB and the Symbolic Math Toolbox to input and form LTI objects in polynomial and factored form for the following frequency functions: [Section: 2.3]

Symbolic Math SM

a. $G(s) = \frac{45(s^2+37s+74)(s^3+28s^2+32s+16)}{(s+39)(s+47)(s^2+2s+100)(s^3+27s^2+18s+15)}$

b. $G(s) = \frac{56(s+14)(s^3+49s^2+62s+53)}{(s^3+81s^2+76s+65)(s^2+88s+33)(s^2+56s+77)}$

16. Find the transfer function, $G(s) = V_o(s)/V_i(s)$, for each network shown in Figure P2.3. [Section: 2.4]

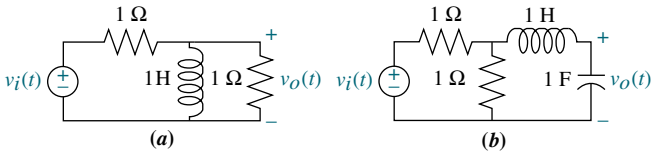


FIGURE P2.3

17. Find the transfer function, $G(s) = V_L(s)/V(s)$, for each network shown in Figure P2.4. [Section: 2.4]

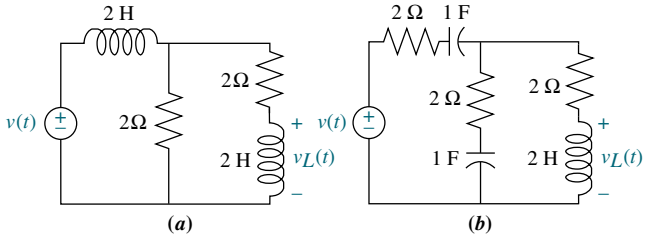


FIGURE P2.4

18. Find the transfer function, $G(s) = V_o(s)/V_i(s)$, for each network shown in Figure P2.5. Solve the problem using mesh analysis. [Section: 2.4]

WileyPLUS WPCS Control Solutions

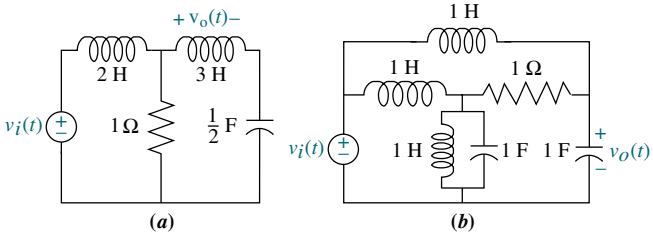


FIGURE P2.5

19. Repeat Problem 18 using nodal equations. [Section: 2.4]

20. a. Write, but do not solve, the mesh and nodal equations for the network of Figure P2.6. [Section: 2.4]

b. Use MATLAB, the Symbolic Math Toolbox, and the equations found in part a to solve for the transfer function, $G(s) = V_o(s)/V(s)$. Use both the

Symbolic Math SM

mesh and nodal equations and show that either set yields the same transfer function.[Section: 2.4]

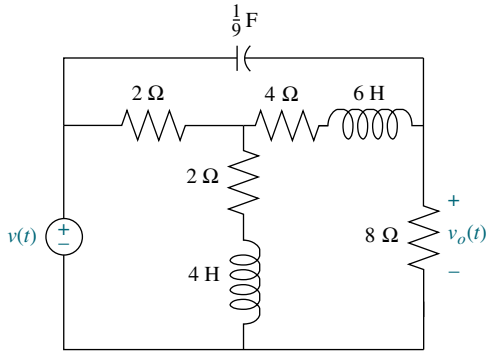
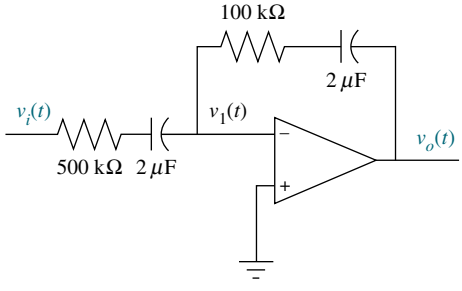
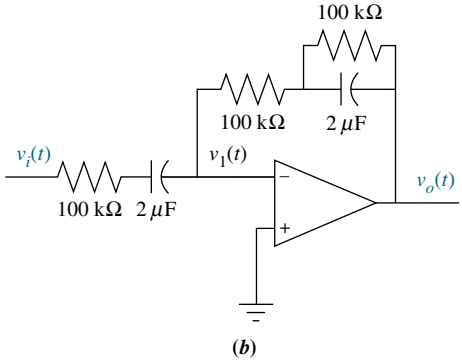


FIGURE P2.6

21. Find the transfer function, $G(s) = V_o(s)/V_i(s)$, for each operational amplifier circuit shown in Figure P2.7. [Section: 2.4]



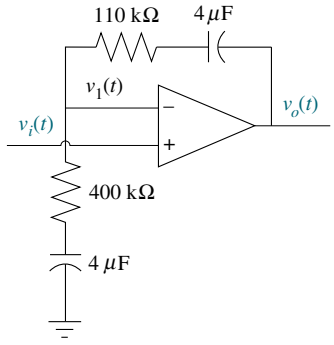
(*a*)



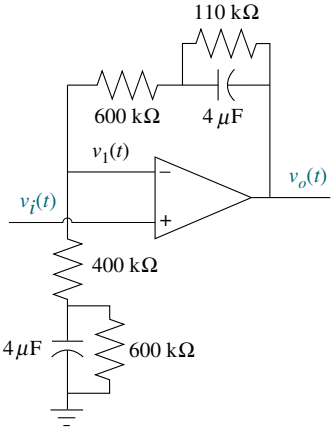
(*b*)

FIGURE P2.7

22. Find the transfer function, $G(s) = V_o(s)/V_i(s)$, for each operational amplifier circuit shown in Figure P2.8. [Section: 2.4]



(*a*)



(*b*)

FIGURE P2.8

23. Find the transfer function, $G(s) = X_1(s)/F(s)$, for the translational mechanical system shown in Figure P2.9. [Section: 2.5]

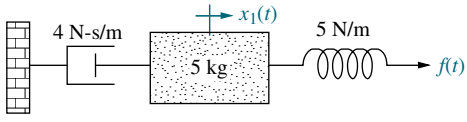


FIGURE P2.9

24. Find the transfer function, $G(s) = X_2(s)/F(s)$, for the translational mechanical network shown in Figure P2.10. [Section: 2.5]

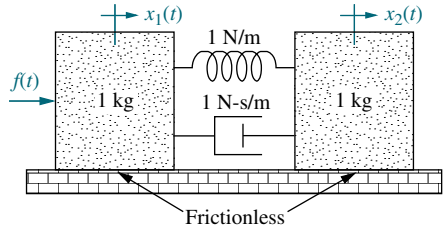


FIGURE P2.10

25. Find the transfer function, $G(s) = X_2(s)/F(s)$, for the translational mechanical system shown in Figure P2.11. (Hint: place a zero mass at $x_2(t)$.) [Section: 2.5]

WileyPLUS WPCS Control Solutions

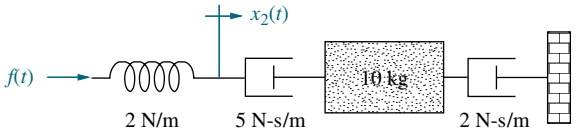


FIGURE P2.11

26. For the system of Figure P2.12 find the transfer function, $G(s) = X_1(s)/F(s)$. [Section: 2.5]

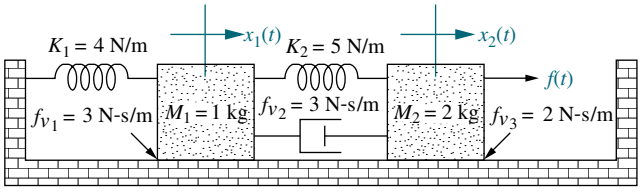


FIGURE P2.12

27. Find the transfer function, $G(s) = X_3(s)/F(s)$, for the translational mechanical system shown in Figure P2.13. [Section: 2.5]

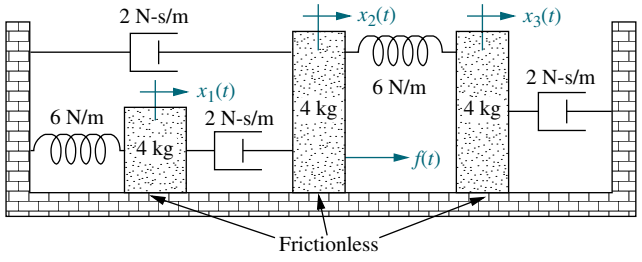
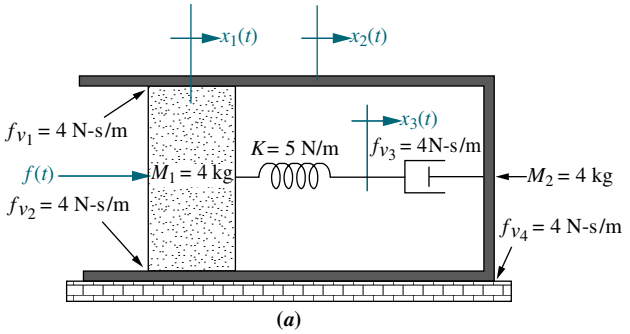
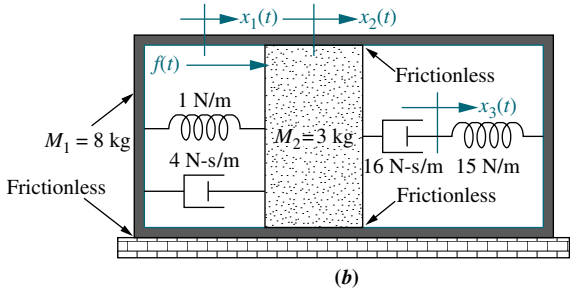


FIGURE P2.13

28. Find the transfer function, $X_3(s)/F(s)$, for each system shown in Figure P2.14. [Section: 2.5]



(*a*)



(*b*)

FIGURE P2.14

29. Write, but do not solve, the equations of motion for the translational mechanical system shown in Figure P2.15. [Section: 2.5]

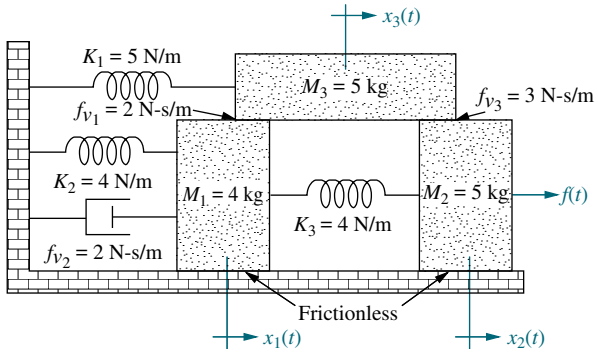
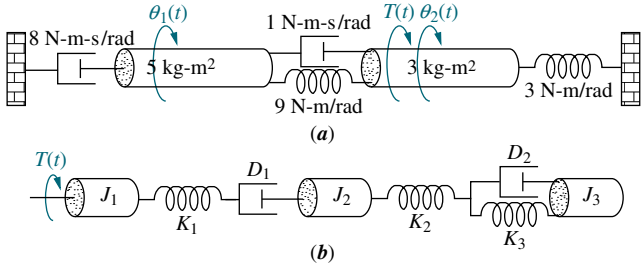


FIGURE P2.15

30. For each of the rotational mechanical systems shown in Figure P2.16, write, but do not solve, the equations of motion. [Section: 2.6]



(*a*)

(*b*)

FIGURE P2.16

31. For the rotational mechanical system shown in Figure P2.17, find the transfer function $G(s) = \theta_2(s)/T(s)$ [Section: 2.6]

WileyPLUS WPCS Control Solutions

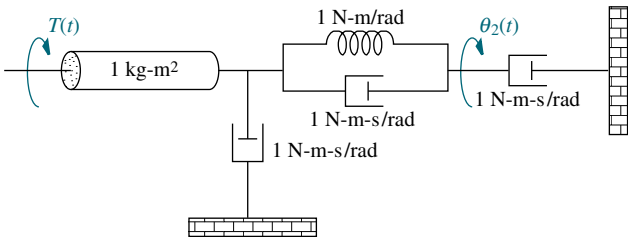


FIGURE P2.17

32. For the rotational mechanical system with gears shown in Figure P2.18, find the transfer function, $G(s) = \theta_3(s)/T(s)$. The gears have inertia and bearing friction as shown. [Section: 2.7]

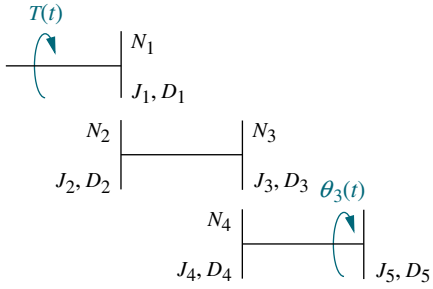


FIGURE P2.18

33. For the rotational system shown in Figure P2.19, find the transfer function, $G(s) = \theta_2(s)/T(s)$. [Section: 2.7]

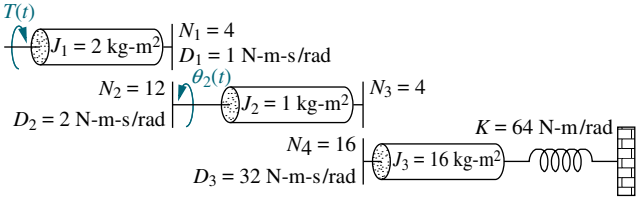


FIGURE P2.19

34. Find the transfer function, $G(s) = \theta_2(s)/T(s)$, for the rotational mechanical system shown in Figure P2.20. [Section: 2.7]

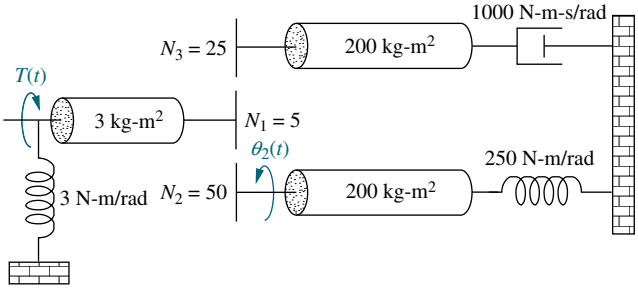


FIGURE P2.20

35. Find the transfer function, $G(s) = \theta_4(s)/T(s)$, for the rotational system shown in Figure P2.21. [Section: 2.7]

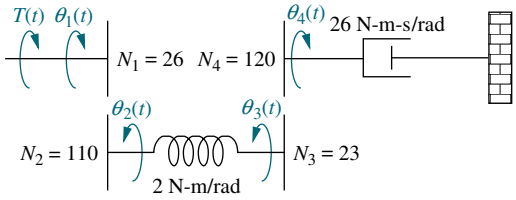


FIGURE P2.21

36. For the rotational system shown in Figure P2.22, find the transfer function, $G(s) = \theta_L(s)/T(s)$. [Section: 2.7]

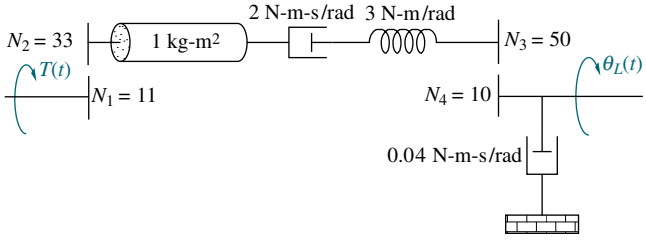


FIGURE P2.22

37. For the rotational system shown in Figure P2.23, write the equations of motion from which the transfer function, $G(s) = \theta_1(s)/T(s)$, can be found. [Section: 2.7]

WileyPLUS WPCS Control Solutions

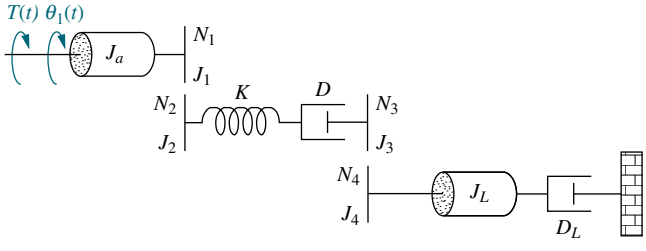


FIGURE P2.23

38. Given the rotational system shown in Figure P2.24, find the transfer function, $G(s) = \theta_6(s)/\theta_1(s)$. [Section: 2.7]

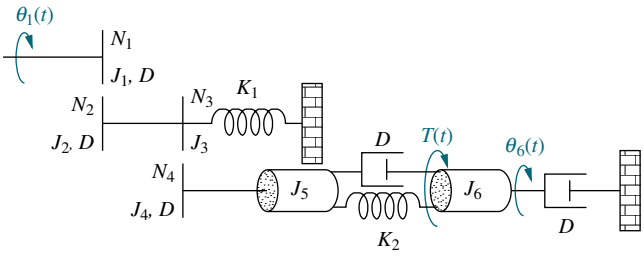


FIGURE P2.24

39. In the system shown in Figure P2.25, the inertia, J, of radius, r, is constrained to move only about the stationary axis A. A viscous damping force of translational value f_v exists between the bodies J and M. If an external force, $f(t)$, is applied to the mass, find the transfer function, $G(s) = \theta(s)/F(s)$. [Sections: 2.5; 2.6]

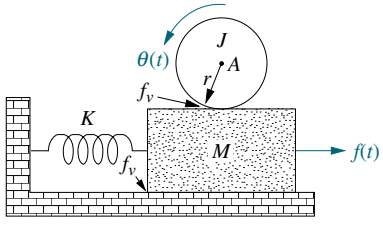


FIGURE P2.25

40. For the combined translational and rotational system shown in Figure P2.26, find the transfer function, $G(s) = X(s)/T(s)$. [Sections: 2.5; 2.6; 2.7]

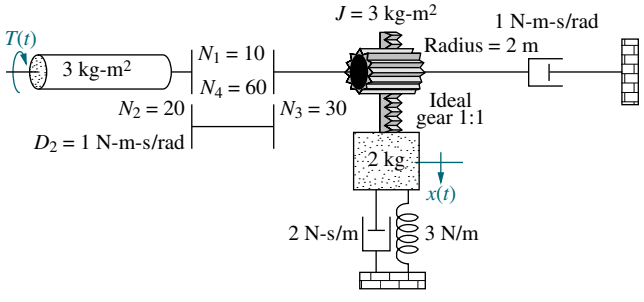


FIGURE P2.26

41. Given the combined translational and rotational system shown in Figure P2.27, find the transfer function, $G(s) = X(s)/T(s)$. [Sections: 2.5; 2.6]

WileyPLUS
WPCS
Control Solutions

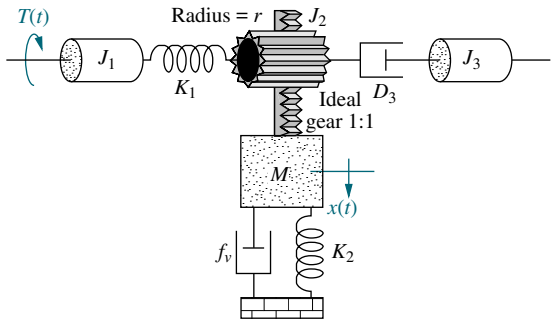


FIGURE P2.27

42. For the motor, load, and torque-speed curve shown in Figure P2.28, find the transfer function, $G(s) = \theta_L(s)/E_a(s)$. [Section: 2.8]

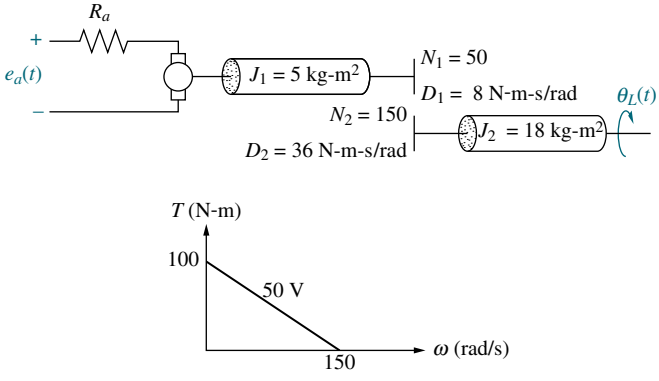


FIGURE P2.28

43. The motor whose torque-speed characteristics are shown in Figure P2.29 drives the load shown in the diagram. Some of the gears have inertia. Find the transfer function, $G(s) = \theta_2(s)/E_a(s)$. [Section: 2.8]

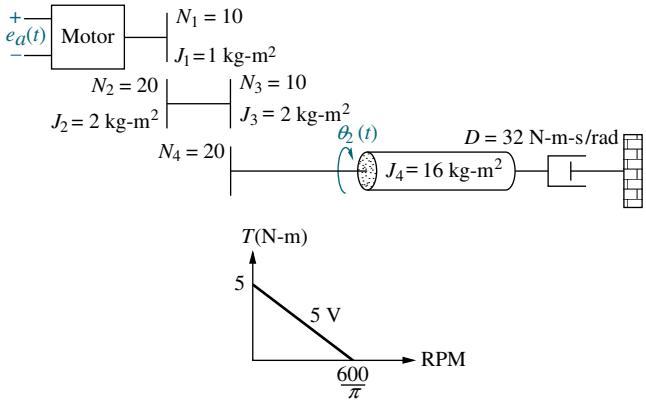


FIGURE P2.29

44. A dc motor develops 55 N-m of torque at a speed of 600 rad/s when 12 volts are applied. It stalls out at this voltage with 100 N-m of torque. If the inertia and damping of the armature are 7 kg-m^2 and 3 N-m-s/rad, respectively, find the transfer function, $G(s) = \theta_L(s)/E_a(s)$, of this motor if it drives an inertia load of 105 kg-m^2 through a gear train, as shown in Figure P2.30. [Section: 2.8]

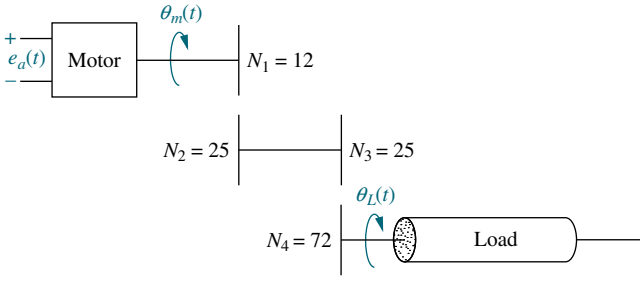


FIGURE P2.30

45. In this chapter, we derived the transfer function of a dc motor relating the angular displacement output to the armature voltage input. Often we want to control the output torque rather than the displacement. Derive the transfer function of the motor that relates output torque to input armature voltage. [Section: 2.8]

WileyPLUS WPCS Control Solutions

46. Find the transfer function, $G(s) = X(s)/E_a(s)$, for the system shown in Figure P2.31. [Sections: 2.5–2.8]

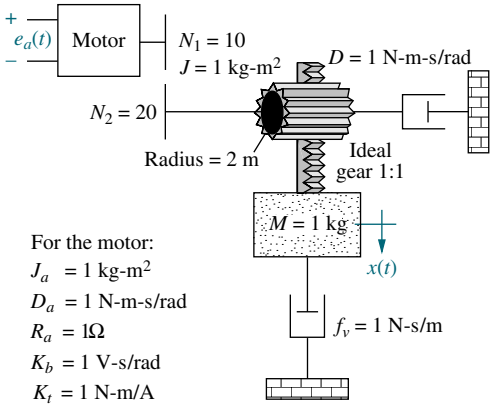


FIGURE P2.31

47. Find the series and parallel analogs for the translational mechanical system shown in Figure 2.20 in the text. [Section: 2.9]

48. Find the series and parallel analogs for the rotational mechanical systems shown in Figure P2.16(b) in the problems. [Section: 2.9]

49. A system's output, c, is related to the system's input, r, by the straight-line relationship, $c = 5r + 7$. Is the system linear? [Section: 2.10]

50. Consider the differential equation

$$\frac{d^2x}{dt^2} + 3\frac{dx}{dt} + 2x = f(x)$$

where $f(x)$ is the input and is a function of the output, x. If $f(x) = \sin x$, linearize the differential equation for small excursions. [Section: 2.10]

a. $x = 0$

b. $x = \pi$

51. Consider the differential equation

WileyPLUS WPCS Control Solutions

$$\frac{d^3x}{dt^3} + 10\frac{d^2x}{dt^2} + 31\frac{dx}{dt} + 30x = f(x)$$

where $f(x)$ is the input and is a function of the output, x. If $f(x) = e^{-x}$, linearize the differential equation for x near 0. [Section: 2.10]

52. Many systems are *piecewise* linear. That is, over a *large* range of variable values, the system can be described linearly. A system with amplifier saturation is one such example. Given the differential equation

$$\frac{d^2x}{dt^2} + 17\frac{dx}{dt} + 50x = f(x)$$

assume that $f(x)$ is as shown in Figure P2.32. Write the differential equation for each of the following ranges of x: [Section: 2.10]

a. $-\infty < x < -3$

b. $-3 < x < 3$

c. $3 < x < \infty$

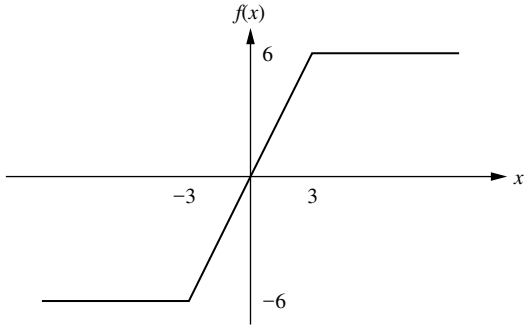


FIGURE P2.32

53. For the translational mechanical system with a nonlinear spring shown in Figure P2.33, find the transfer function, $G(s) = X(s)/F(s)$, for small excursions around $f(t) = 1$. The spring is defined by $x_s(t) = 1 - e^{-f_s(t)}$, where $x_s(t)$ is the spring displacement and $f_s(t)$ is the spring force. [Section: 2.10]

WileyPLUS WPCS Control Solutions

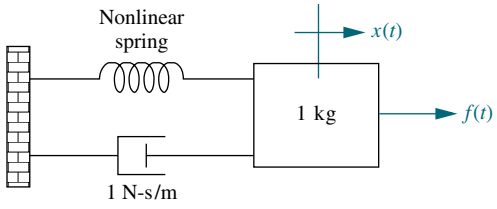


FIGURE P2.33

54. Consider the restaurant plate dispenser shown in Figure P2.34, which consists of a vertical stack of dishes supported by a compressed spring. As each plate is removed, the reduced weight on the dispenser causes the remaining plates to rise. Assume that the mass of the system minus the top plate is M, the viscous friction between the piston and the sides of the cylinder is f_v, the spring constant is K, and the weight of a single plate is W_D. Find the transfer function, $Y(s)/F(s)$, where $F(s)$ is the step reduction in force felt when the top plate is removed, and $Y(s)$ is the vertical displacement of the dispenser in an upward direction.

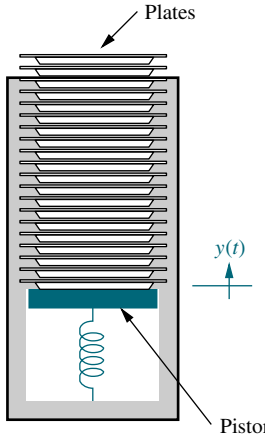


FIGURE P2.34 Plate dispenser

55. Each inner ear in a human has a set of three nearly perpendicular semicircular canals of about 0.28 mm in diameter filled with fluid. Hair-cell transducers that deflect with skull movements and whose main purpose is to work as attitude sensors as well as help us maintain our sense of direction and equilibrium are attached to the canals. As the hair cells move, they deflect a waterproof flap called the *cupula.* It has been shown that the skull and cupula movements are related by the following equation (*Milsum, 1966*):

$$J\ddot{\phi} + b\dot{\phi} + k\phi = (aJ)\ddot{\psi}$$

where

J = moment of inertia of the fluid in the thin tube (constant)
b = torque per unit relative angular velocity (constant)
k = torque per unit relative angular displacement (constant)
a = constant
$\phi(t)$ = angular deflection of the cupula (output)
$\ddot{\psi}(t)$ = skull's angular acceleration (input)

Find the transfer function $\dfrac{\Phi(s)}{\ddot{\Psi}(s)}$.

56. Diabetes is an illness that has risen to epidemic proportions, affecting about 3% of the total world population in 2003. A differential equation model that describes the total population size of diabetics is

$$\frac{dC(t)}{dt} = -(\lambda + \mu + \delta + \gamma + \nu)C(t) + \lambda N(t)$$

$$\frac{dN(t)}{dt} = -(\nu + \delta)C(t) - \mu N(t) + I(t)$$

with the initial conditions $C(0) = C_0$ and $N(0) = N_0$ and

$I(t)$ = the system input: the number of new cases of diabetes
$C(t)$ = number of diabetics with complications
$N(t)$ = the system output: the total number of diabetics with and without complications
μ = natural mortality rate (constant)
λ = probability of developing a complication (constant)
δ = mortality rate due to complications (constant)
ν = rate at which patients with complications become severely disabled (constant)
γ = rate at which complications are cured (constant)

Assume the following values for parameters: $\nu = \delta = 0.05/\text{yr}$, $\mu = 0.02/\text{yr}$, $\gamma = 0.08/\text{yr}$, $\lambda = 0.7$, with initial conditions $C_0 = 47{,}000{,}500$ and $N_0 =$, 61,100,500. Assume also that diabetic incidence is constant $I(t) = I = 6 \times 10^6$ (*Boutayeb, 2004*).

a. Draw a block diagram of the system showing the output $N(s)$, the input $I(s)$, the transfer function, and the initial conditions.

b. Use any method to find the analytic expression for $N(t)$ for $t \geq 0$.

57. The circuit shown in Figure P2.35(a) is excited with the pulse shown in Figure P2.35(b).

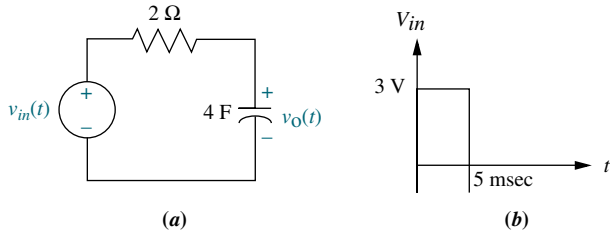


FIGURE P2.35

The Laplace transform can be used to calculate $v_o(t)$ in two different ways: The "exact" method is performed by writing $v_i(t) = 3[u(t) - u(t - 0.005)]$, from which we use the Laplace transform to obtain

$$V_{in}(s) = 3\frac{1 - e^{-0.005s}}{s}$$

(Hint: look at Item 5 in Table 2.2, the time shift theorem.) In the second approach the pulse is approximated by an impulse input having the same area (energy) as the original input. From Figure P2.35(b): $v_{in}(t) \approx (3\,\text{V})(5\,\text{msec})\ \delta(t) = 0.015\delta(t)$. In this case, $V_{in}(s) = 0.015$. This approximation can be used as long as the width of the pulse of Figure P2.35(b) is much smaller than the circuit's smallest time constant. (Here, $\tau = RC = (2\Omega)(4\,\text{F}) = 8\,\text{sec} \gg 5\,\text{msec}$.)

a. Assuming the capacitor is initially discharged, obtain an analytic expression for $v_o(t)$ using both methods.

b. Plot the results of both methods using any means available to you, and compare both outputs. Discuss the differences.

58. In a magnetic levitation experiment a metallic object is held up in the air suspended under an electromagnet. The vertical displacement of the object can be described by the following nonlinear differential equation (*Galvão, 2003*):

$$m\frac{d^2H}{dt^2} = mg - k\frac{I^2}{H^2}$$

where

m = mass of the metallic object
g = gravity acceleration constant
k = a positive constant
H = distance between the electromagnet and the metallic object (output signal)
I = electromagnet's current (input signal)

a. Show that a system's equilibrium will be achieved when $H_0 = I_0\sqrt{\frac{k}{mg}}$.

b. Linearize the equation about the equilibrium point found in Part **a** and show that the resulting transfer function obtained from the linearized differential equation can be expressed as

$$\frac{\delta H(s)}{\delta I(s)} = -\frac{a}{s^2 - b^2}$$

with $a > 0$. Hint: to perform the linearization, define $\delta H = H(t) - H_0$ and $\delta I = I(t) - I_0$; substitute into the original equation. This will give

$$m\frac{d^2(H_0 + \delta H)}{dt^2} = mg - k\frac{(I_0 + \delta I)^2}{(H_0 + \delta H)^2} = \gamma$$

Now get a first-order Taylor's series approximation on the right-hand side of the equation. Namely, calculate

$$m\frac{d^2\delta H}{dt^2} = \frac{\partial\gamma}{\partial\delta H}\bigg|_{\delta H=0,\ \delta I=0}\delta H + \frac{\partial\gamma}{\partial\delta I}\bigg|_{\delta H=0,\ \delta I=0}\delta I$$

59. Figure P2.36 shows a quarter-car model commonly used for analyzing suspension systems. The car's tire is considered to act as a spring without damping, as shown. The parameters of the model are (*Lin, 1997*)

M_b = car's body mass
M_{us} = wheel's mass
K_a = strut's spring constant
K_t = tire's spring constant
f_v = strut's damping constant
r = road disturbance (input)
x_s = car's vertical displacement
x_w = wheel's vertical displacement

Obtain the transfer function from the road disturbance to the car's vertical displacement $\frac{X_s(s)}{R(s)}$.

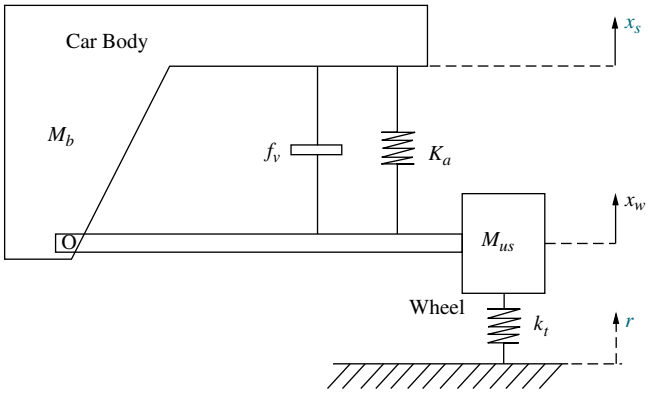


FIGURE P2.36 Quarter-car model used for suspension design. (© 1997 IEEE)

60. Enzymes are large proteins that biological systems use to increase the rate at which reactions occur. For example, food is usually composed of large molecules that are hard to digest; enzymes break down the large molecules into small nutrients as part of the digestive process. One such enzyme is amylase, contained in human saliva. It is commonly known that if you place a piece of uncooked pasta in your mouth its taste will change from paper-like to sweet as amylase breaks down the carbohydrates into sugars. Enzyme breakdown is often expressed by the following relation:

$$S + E \underset{k_1}{\overset{k_s}{\rightleftarrows}} C \overset{k_1}{\rightarrow} P$$

In this expression a substrate (S) interacts with an enzyme (E) to form a combined product (C) at a rate k_1. The intermediate compound is reversible and gets disassociated at a rate k_{-1}. Simultaneously some of the compound is transformed into the final product (P) at a rate k_2. The kinetics describing this reaction are known as the Michaelis-Menten equations and consist of four nonlinear differential equations. However, under some conditions these equations can be simplified. Let E_0 and S_0 be the initial concentrations of enzyme and substrate, respectively. It is generally accepted that under some energetic conditions or when the enzyme concentration is very big ($E_0 \gg S_0$), the kinetics for this reaction are given by

$$\frac{dS}{dt} = k_\psi(\widetilde{K}_s C - S)$$
$$\frac{dC}{dt} = k_\psi(S - \widetilde{K}_M C)$$
$$\frac{dP}{dt} = k_2 C$$

where the following constant terms are used (*Schnell, 2004*) :

$$k_\psi = k_1 E_0$$
$$\widetilde{K}_s = \frac{k-1}{k_\psi}$$

and

$$\widetilde{K}_M = \widetilde{K}_s + \frac{k_2}{k_\psi}$$

a. Assuming the initial conditions for the reaction are $S(0) = S_0$, $E(0) = E_0$, $C(0) = P(0) = 0$, find the Laplace transform expressions for S, C, and P: $\mathscr{L}\{S\}$, $\mathscr{L}\{C\}$, and $\mathscr{L}\{P\}$, respectively.

b. Use the final theorem to find $S(\infty)$, $C(\infty)$, and $P(\infty)$.

61. Humans are able to stand on two legs through a complex feedback system that includes several sensory inputs—equilibrium and visual along with muscle actuation. In order to gain a better understanding of the workings of the postural feedback mechanism, an individual is asked to stand on a platform to which sensors are attached at the base. Vibration actuators are attached with straps to the individual's calves. As the vibration actuators are stimulated, the individual sways and movements are recorded. It was hypothesized that the human postural dynamics are analogous to those of a cart with a balancing standing pole attached (inverted pendulum). In that case, the dynamics can be described by the following two equations:

$$J\frac{d^2\theta}{dt^2} = mgl\sin\theta(t) + T_{\text{bal}} + T_d(t)$$
$$T_{\text{bal}}(t) = -mgl\sin\theta(t) + kJ\theta(t) - \eta J\dot{\theta}(t) - \rho J\int_0^t \theta(t)dt$$

where m is the individual's mass; l is the height of the individual's center of gravity; g is the gravitational constant; J is the individual's equivalent moment of inertia; η, ρ, and k are constants given by the body's postural control system; $\theta(t)$ is the

individual's angle with respect to a vertical line; $T_{\text{bal}}(t)$ is the torque generated by the body muscles to maintain balance; and $T_d(t)$ is the external torque input disturbance. Find the transfer function $\dfrac{\Theta(s)}{T_d(s)}$ (*Johansson, 1988*).

62. Figure P2.37 shows a crane hoisting a load. Although the actual system's model is highly nonlinear, if the rope is considered to be stiff with a fixed length L, the system can be modeled using the following equations:

$$\begin{aligned} m_L\ddot{x}_{La} &= m_L g\phi \\ m_T\ddot{x}_T &= f_T - m_L g\phi \\ x_{La} &= x_T - x_L \\ x_L &= L\phi \end{aligned}$$

where m_L is the mass of the load, m_T is the mass of the cart, x_T and x_L are displacements as defined in the figure, ϕ is the rope angle with respect to the vertical, and f_T is the force applied to the cart (*Marttinen, 1990*).

a. Obtain the transfer function from cart velocity to rope angle $\dfrac{\Phi(s)}{V_T(s)}$.

b. Assume that the cart is driven at a constant velocity V_0 and obtain an expression for the resulting $\phi(t)$. Show that under this condition, the load will sway with a frequency $\omega_0 = \sqrt{\dfrac{g}{L}}$.

c. Find the transfer function from the applied force to the cart's position, $\dfrac{X_T(s)}{F_T(s)}$.

d. Show that if a constant force is applied to the cart, its velocity will increase without bound as $t \to \infty$.

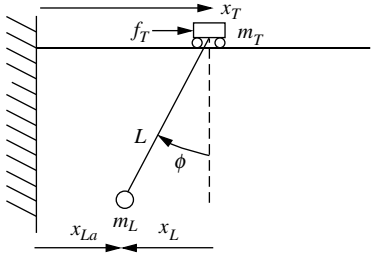


FIGURE P2.37 (© 1990 IEEE)

63. In 1978, Malthus developed a model for human growth population that is also commonly used to model bacterial growth as follows. Let $N(t)$ be the population density observed at time t. Let K be the rate of reproduction per unit time. Neglecting population deaths, the population density at a time $t + \Delta t$ (with small Δt) is given by

$$N(t + \Delta t) \approx N(t) + KN(t)\Delta t$$

which also can be written as

$$\frac{N(t + \Delta t) - N(t)}{\Delta t} = KN(t)$$

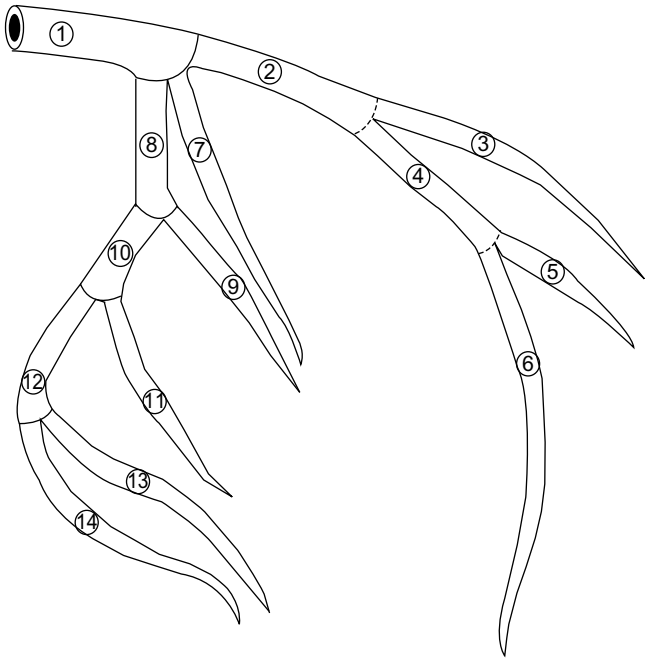
Since $N(t)$ can be considered to be a very large number, letting $\Delta t \to 0$ gives the following differential equation (*Edelstein-Keshet, 2005*):

$$\frac{dN(t)}{dt} = KN(t)$$

a. Assuming an initial population $N(0) = N_0$, solve the differential equation by finding $N(t)$.

b. Find the time at which the population is double the initial population.

64. Blood vessel blockages can in some instances be diagnosed through noninvasive techniques such as the use of sensitive microphones to detect flow acoustic anomalies. In order to predict the sound properties of the left coronary artery, a model has been developed that partitions the artery into 14 segments, as shown in Figure P2.38(*a*).



(*a*)

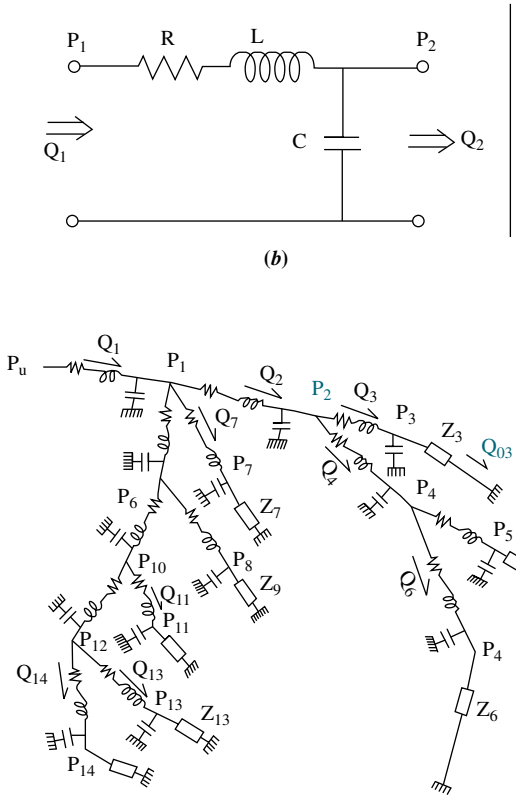


FIGURE P2.38 (© 1990 IEEE)

Each segment is then modeled through the analogous electrical circuit of Figure P2.38(b), resulting in the total model shown in Figure P2.38(c), where eight terminal resistances (Z) have been added. In the electrical model, pressure is analogous to voltage and blood flow is analogous to current. As an example, for Segment 3 it was experimentally verified that $R_3 = 4176\,\Omega$, $C_3 = 0.98\,\mu$F, $L_3 = 140.6$ H, and $Z_3 = 308,163\,\Omega$ (*Wang, 1990*).

a. For Segment 3, find the transfer function from input pressure to blood flow through Z_3, $\dfrac{Q_{03}(s)}{P_2(s)}$.

b. It is well known in circuit analysis that if a constant input is applied to a circuit such as the one of Figure P2.38(b), the capacitor can be substituted by an open circuit and the inductor can be substituted by a short circuit as time approaches infinity. Use this fact to calculate the flow through Z_3 after a constant unit pressure pulse is applied and time approaches infinity.

c. Verify the result obtained in Part **b** using the transfer function obtained in Part **a** and applying the final value theorem.

65. In order to design an underwater vehicle that has the characteristics of both a long-range transit vehicle (torpedo-like) and a highly maneuverable low-speed vehicle (boxlike), researchers have developed a thruster that mimics that of squid jet locomotion (*Krieg, 2008*). It has been demonstrated there that the average normalized thrust due to a command step input, $U(s) = \dfrac{T_{ref}}{s}$, is given by:

$$T(t) = T_{ref}(1 - e^{-\lambda t}) + a\sin(2\pi f t)$$

where T_{ref} is the reference or desired thrust, λ is the system's damping constant, a is the amplitude of the oscillation caused by the pumping action of the actuator, f is the actuator frequency, and $T(t)$ is the average resulting normalized thrust. Find the thruster's transfer function $\dfrac{T(s)}{U(s)}$. Show all steps.

66. The Gompertz growth model is commonly used to model tumor cell growth. Let $v(t)$ be the tumor's volume, then

$$\frac{dv(t)}{dt} = \lambda e^{-\alpha t} v(t)$$

where λ and α are two appropriate constants (*Edelstein-Keshet, 2005*).

a. Verify that the solution to this equation is given by $v(t) = v_0 e^{\lambda/\alpha(1-e^{-\alpha t})}$, where v_0 is the initial tumor volume.

b. This model takes into account the fact that when nutrients and oxygen are scarce at the tumor's core, its growth is impaired. Find the final predicted tumor volume (let $t \to \infty$).

c. For a specific mouse tumor, it was experimentally found that $\lambda = 2.5$ days, $\alpha = 0.1$ days with $v_0 = 50 \times 10^{-3}$ mm^3 (*Chignola, 2005*). Use any method available to make a plot of $v(t)$ vs. t.

d. Check the result obtained in Part **b** with the results from the graph from Part **c.**

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

67. High-speed rail pantograph. Problem 21 in Chapter 1 discusses active control of a pantograph mechanism for high-speed rail systems. The diagram for the pantograph and catenary coupling is shown in Figure P2.39(a). Assume the simplified model shown in Figure P2.39(b), where the catenary is represented by the spring, K_{ave} (*O'Connor, 1997*).

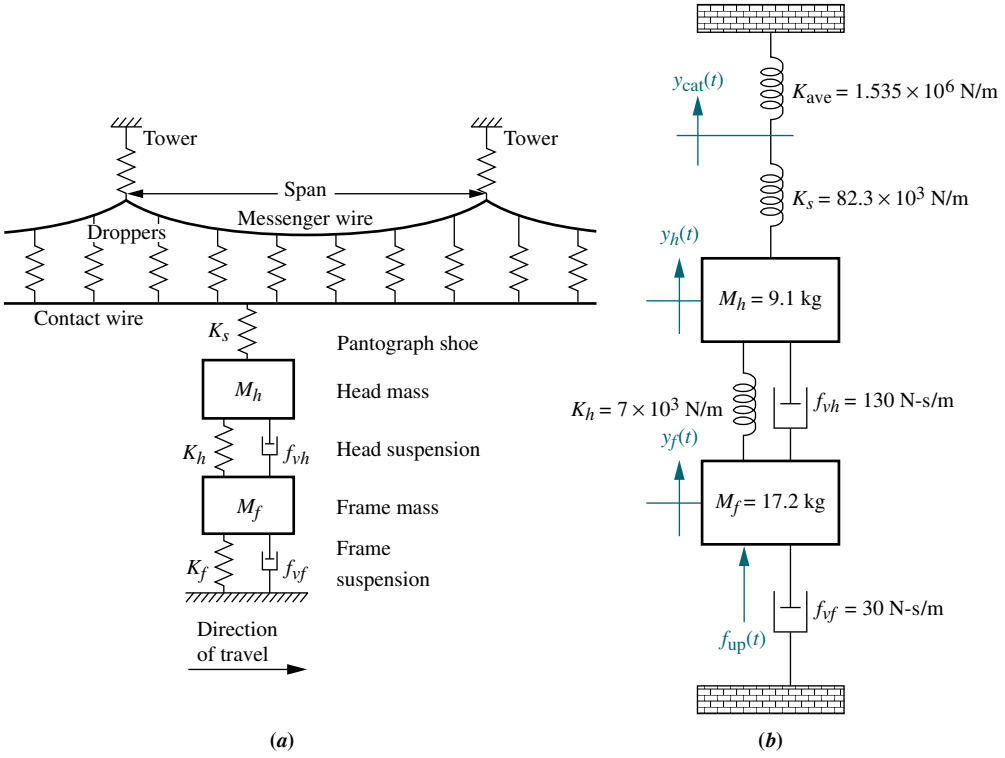


FIGURE P2.39 **a.** Coupling of pantograph and catenary; **b.** simplified representation showing the active-control force (Reprinted with permission of ASME.)

a. Find the transfer function, $G_1(s) = Y_{cat}(s)/F_{up}(s)$, where $y_{cat}(t)$ is the catenary displacement and $f_{up}(t)$ is the upward force applied to the pantograph under active control.

b. Find the transfer function $G_2(s) = Y_h(s)/F_{up}(s)$, where $y_h(t)$ is the pantograph head displacement.

c. Find the transfer function, $G(s) = (Y_h(s) - Y_{cat}(s))/F_{up}(s)$.

68. Control of HIV/AIDS. HIV inflicts its damage by infecting healthy CD4 + T cells (a type of white blood cell) that are necessary to fight infection. As the virus embeds in a T cell and the immune system produces more of these cells to fight the infection, the virus propagates in an opportunistic fashion. As we now develop a simple HIV model, refer to Figure P2.40. Normally T cells are produced at a rate s and die at a rate d. The HIV virus is present in the bloodstream in the infected individual. These viruses in the bloodstream, called *free viruses*, infect healthy T cells at a rate β. Also, the viruses reproduce through the T cell multiplication process or otherwise at a rate k. Free viruses die at a rate c. Infected T cells die at a rate μ.

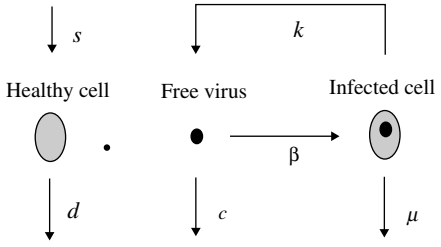


FIGURE P2.40 (© 2004 IEEE)

A simple mathematical model that illustrates these interactions is given by the following equations (*Craig, 2004*):

$$\frac{dT}{dt} = s - dT - \beta Tv$$

$$\frac{dT^*}{dt} = \beta Tv - \mu T^*$$

$$\frac{dv}{dt} = kT^* - cv$$

where

$$T = \text{number of healthy T cells}$$
$$T^* = \text{number of infected T cells}$$
$$v = \text{number of free viruses}$$

a. The system is nonlinear; thus linearization is necessary to find transfer functions as you will do in subsequent chapters. The nonlinear nature of this model can be seen from the above equations. Determine which of these equations are linear, which are nonlinear, and explain why.

b. The system has two equilibrium points. Show that these are given by

$$(T_0,\ T_0^*,\ v_0) = \left(\frac{s}{d},\ 0,\ 0\right)$$

and

$$(T_0,\ T_0^*,\ v_0) = \left(\frac{c\mu}{\beta k},\ \frac{s}{\mu} - \frac{cd}{\beta k},\ \frac{sk}{c\mu} - \frac{d}{\beta}\right)$$

69. Hybrid vehicle. Problem 23 in Chapter 1 discusses the cruise control of serial, parallel, and split-power hybrid electric vehicles (HEVs). The functional block diagrams developed for these HEVs indicated that the speed of a vehicle depends upon the balance between the motive forces (developed by the gasoline engine and/or the electric motor) and running resistive forces. The resistive forces include the aerodynamic drag, rolling resistance, and climbing resistance. Figure P2.41 illustrates the running resistances for a car moving uphill (*Bosch, 2007*).

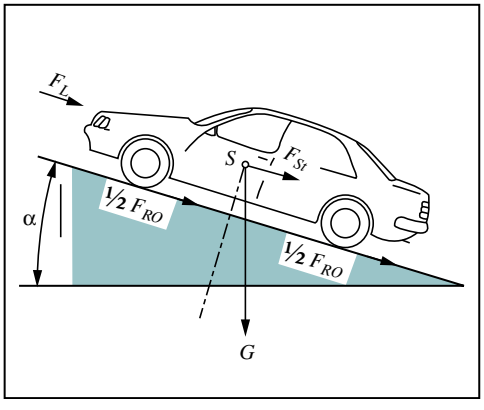


FIGURE P2.41 Running resistances

The total running resistance, F_w, is calculated as $F_w = F_{Ro} + F_L + F_{St}$, where F_{Ro} is the rolling resistance, F_L is the aerodynamic drag, and F_{St} is the climbing resistance. The aerodynamic drag is proportional to the square of the sum of car velocity, v, and the head-wind velocity, v_{hw}, or $v + v_{hw}$. The other two resistances are functions of car weight, G, and the gradient of the road (given by the gradient angle, α), as seen from the following equations:

$$F_{Ro} = fG\cos\alpha = fmg\cos\alpha$$

where

$$f = \text{coefficient of rolling resistance,}$$
$$m = \text{car mass, in kg,}$$
$$g = \text{gravitational acceleration, in m/s}^2.$$

$$F_L = 0.5\rho C_w A(v + v_{hw})^2$$

where

$$\rho = \text{air density, in kg/m}^3,$$
$$C_w = \text{coefficient of aerodynamic drag,}$$
$$A = \text{largest cross-section of the car, in kg/m}^2.$$

$$F_{St} = G\sin\alpha = mg\sin\alpha$$

The motive force, F, available at the drive wheels is:

$$F = \frac{Ti_{tot}}{r}\eta_{tot} = \frac{P\eta_{tot}}{v}$$

where

$$T = \text{motive torque,}$$
$$P = \text{motive power,}$$
$$i_{tot} = \text{total transmission ratio,}$$
$$r = \text{tire radius,}$$
$$\eta_{tot} = \text{total drive-train efficiency.}$$

The surplus force, $F - F_w$, accelerates the vehicle (or retards it when $F_w > F$). Letting $a = \dfrac{F - F_w}{k_m \cdot m}$, where a is the acceleration and k_m is a coefficient that compensates for the apparent increase in vehicle mass due to rotating masses (wheels, flywheel, crankshaft, etc.):

a. Show that car acceleration,[16] a, may be determined from the equation:

$$F = fmg\cos\alpha + mg\sin\alpha + 0.5\rho C_w A(v + v_{hw})^2 + k_m ma$$

[16] Other quantities, such as top speed, climbing ability, etc., may also be calculated by manipulation from that equation.

b. Assuming constant acceleration and using the average value for speed, find the average motive force, F_{av} (in N), and power, P_{av} (in kW) the car needs to accelerate from 40 to 60 km/h in 4 seconds on a level road, ($\alpha = 0°$), under windless conditions, where $v_{hw} = 0$. You are given the following parameters: m = 1590 kg, A = 2 m^2, $f = 0.011$, $\rho = 1.2$ kg/m^3, $C_w = 0.3$, $\eta_{tot} = 0.9$, $k_m = 1.2$. Furthermore, calculate the additional power, P_{add}, the car needs after reaching 60 km/h to maintain its speed while climbing a hill with a gradient $\alpha = 5°$.

c. The equation derived in Part **a** describes the nonlinear car motion dynamics where $F(t)$ is the input to the system, and $v(t)$ the resulting output. Given that the aerodynamic drag is proportional to v^2 under windless conditions, linearize the resulting equation of motion around an average speed, $v_o = 50$ km/h, when the car travels on a level road,[17] where $\alpha = 0°$. (Hint: Expand $v^2 - v_0^2$ in a truncated Taylor series). Write that equation of motion and represent it with a block diagram in which the block G_v represents the vehicle dynamics. The output of that block is the car speed, $v(t)$, and the input is the excess motive force, $F_e(t)$, defined as: $F_e = F - F_{St} - F_{Ro} + F_o$, where F_o the constant component of the linearized aerodynamic drag.

d. Use the equation in Part **c** to find the vehicle transfer function: $G_v(s) = V(s)/F_e(s)$.

[17] Note that on a level road the climbing resistance is $F_{St} = 0$, since $\sin\alpha = \sin 0° = 0$.

Cyber Exploration Laboratory

Experiment 2.1

Objectives To learn to use MATLAB to (1) generate polynomials, (2) manipulate polynomials, (3) generate transfer functions, (4) manipulate transfer functions, and (5) perform partial-fraction expansions.

Minimum Required Software Packages MATLAB and the Control System Toolbox

Prelab

1. Calculate the following by hand or with a calculator:
   - **a.** The roots of $P_1 = s^6 + 7s^5 + 2s^4 + 9s^3 + 10s^2 + 12s + 15$
   - **b.** The roots of $P_2 = s^6 + 9s^5 + 8s^4 + 9s^3 + 12s^2 + 15s + 20$
   - **c.** $P_3 = P_1 + P_2$; $P_4 = P_1 - P_2$; $P_5 = P_1 P_2$
2. Calculate by hand or with a calculator the polynomial
$$P_6 = (s+7)(s+8)(s+3)(s+5)(s+9)(s+10)$$
3. Calculate by hand or with a calculator the following transfer functions:
   - **a.** $G_1(s) = \dfrac{20(s+2)(s+3)(s+6)(s+8)}{s(s+7)(s+9)(s+10)(s+15)}$,
     represented as a numerator polynomial divided by a denominator polynomial.
   - **b.** $G_2(s) = \dfrac{s^4 + 17s^3 + 99s^2 + 223s + 140}{s^5 + 32s^4 + 363s^3 + 2092s^2 + 5052s + 4320}$,
     expressed as factors in the numerator divided by factors in the denominator, similar to the form of $G_1(s)$ in Prelab 3**a.**
   - **c.** $G_3(s) = G_1(s) + G_2(s)$; $G_4(s) = G_1(s) - G_2(s)$; $G_5(s) = G_1(s)G_2(s)$
     expressed as factors divided by factors and expressed as polynomials divided by polynomials.

4. Calculate by hand or with a calculator the partial-fraction expansion of the following transfer functions:

   a. $G_6 = \dfrac{5(s+2)}{s(s^2+8s+15)}$

   b. $G_7 = \dfrac{5(s+2)}{s(s^2+6s+9)}$

   c. $G_8 = \dfrac{5(s+2)}{s(s^2+6s+34)}$

Lab

1. Use MATLAB to find P_3, P_4, and P_5 in Prelab 1.
2. Use only one MATLAB command to find P_6 in Prelab 2.
3. Use only two MATLAB commands to find $G_1(s)$ in Prelab 3**a** represented as a polynomial divided by a polynomial.
4. Use only two MATLAB commands to find $G_2(s)$ expressed as factors in the numerator divided by factors in the denominator.
5. Using various combinations of $G_1(s)$ and $G_2(s)$, find $G_3(s)$, $G_4(s)$, and $G_5(s)$. Various combinations implies mixing and matching $G_1(s)$ and $G_2(s)$ expressed as factors and polynomials. For example, in finding $G_3(s)$, $G_1(s)$ can be expressed in factored form and $G_2(s)$ can be expressed in polynomial form. Another combination is $G_1(s)$ and $G_2(s)$ both expressed as polynomials. Still another combination is $G_1(s)$and $G_2(s)$ both expressed in factored form.
6. Use MATLAB to evaluate the partial fraction expansions shown in Prelab 4.

Postlab

1. Discuss your findings for Lab 5. What can you conclude?
2. Discuss the use of MATLAB to manipulate transfer functions and polynomials. Discuss any shortcomings in using MATLAB to evaluate partial fraction expansions.

Experiment 2.2

Objectives To learn to use MATLAB and the Symbolic Math Toolbox to (1) find Laplace transforms for time functions, (2) find time functions from Laplace transforms, (3) create LTI transfer functions from symbolic transfer functions, and (4) perform solutions of symbolic simultaneous equations.

Minimum Required Software Packages MATLAB, the Symbolic Math Toolbox, and the Control System Toolbox

Prelab

1. Using a hand calculation, find the Laplace transform of:
   $$f(t) = 0.0075 - 0.00034e^{-2.5t}\cos(22t) + 0.087e^{-2.5t}\sin(22t) - 0.0072e^{-8t}$$
2. Using a hand calculation, find the inverse Laplace transform of
   $$F(s) = \frac{2(s+3)(s+5)(s+7)}{s(s+8)(s^2+10s+100)}$$
3. Use a hand calculation to solve the circuit for the loop currents shown in Figure P2.42.

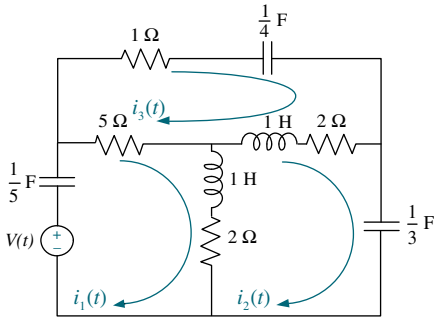


FIGURE P2.42

Lab

1. Use MATLAB and the Symbolic Math Toolbox to
   - **a.** Generate symbolically the time function $f(t)$ shown in Prelab 1.
   - **b.** Generate symbolically $F(s)$ shown in Prelab 2. Obtain your result symbolically in both factored and polynomial forms.
   - **c.** Find the Laplace transform of $f(t)$ shown in Prelab 1.
   - **d.** Find the inverse Laplace transform of $F(s)$ shown in Prelab 2.
   - **e.** Generate an LTI transfer function for your symbolic representation of $F(s)$ in Prelab 2 in both polynomial form and factored form. Start with the $F(s)$ you generated symbolically.
   - **f.** Solve for the loop currents in Prelab 3.

Postlab

1. Discuss the advantages and disadvantages between the Symbolic Math Toolbox and MATLAB alone to convert a transfer function from factored form to polynomial form and vice versa.
2. Discuss the advantages and disadvantages of using the Symbolic Math Toolbox to generate LTI transfer functions.
3. Discuss the advantages of using the Symbolic Math Toolbox to solve simultaneous equations of the type generated by the electrical network in Prelab 3. Is it possible to solve the equations via MATLAB alone? Explain.
4. Discuss any other observations you had using the Symbolic Math Toolbox.

Experiment 2.3

Objective To learn to use LabVIEW to generate and manipulate polynomials and transfer functions.

Minimum Required Software Packages LabVIEW and the LabVIEW Control Design and Simulation Module.

Prelab

1. Study Appendix D, Sections D.1 through Section D.4, Example D.1.
2. Perform by hand the calculations stated in Prelab 1 of Experiment 2.1.
3. Find by a hand calculation the polynomial whose roots are: $-7, -8, -3, -5, -9$, and -10.
4. Perform by hand a partial-fraction expansion of $G(s) = \dfrac{5s + 10}{s^3 + 8s^2 + 15s}$.
5. Find by a hand calculation $G_1(s) + G_2(s)$, $G_1(s) - G_2(s)$, and $G_1(s)G_2(s)$, where $G_1(s) = \dfrac{1}{s^2 + s + 2}$ and $G_2(s) = \dfrac{s + 1}{s^2 + 4s + 3}$.

Lab

1. Open the LabVIEW functions palette and select the **Mathematics/Polynomial** palette.
2. Generate the polynomials enumerated in Prelab 1a and lb of Experiment 2.1.
3. Generate the polynomial operations stated in Prelab 1c of Experiment 2.1.
4. Generate a polynomial whose roots are those stated in Prelab 3 of this experiment.

5. Generate the partial fraction expansion of the transfer function given in Prelab 4 of this experiment.
6. Using the **Control Design and Simulation/Control Design/Model Construction** palette, construct the two transfer functions enumerated in Prelab 5.
7. Using the **Control Design and Simulation/Control Design/Model Interconnection** palette, display the results of the mathematical operations enumerated in Prelab 5 of this experiment.

Postlab

1. Compare the polynomial operations obtained in Lab 3 to those obtained in Prelab 2.
2. Compare the polynomial displayed in Lab 4 with that calculated in Prelab 3.
3. Compare the partial-fraction expansion obtained in Lab 5 with that calculated in Prelab 4.
4. Compare the results of the mathematical operations found in Lab 7 to those calculated in Prelab 5.

Bibliography

Aggarwal, J. K. *Notes on Nonlinear Systems.* Van Nostrand Reinhold, New York, 1972.

Bosch, R. GmbH, *Bosch Automotive Handbook*, 7th ed. John Wiley & Sons Ltd., UK, 2007.

Boutayeb, A., Twizell, E. H., Achouayb, K., and Chetouani, A. Mathematical Model for the Burden of Diabetes and Its Complications. *BioMedical Engineering OnLine*, 2004, Retrieved from http://www.biomedical-engineering-online.com/content/3/1/20, pp. 1–19.

Cannon, R. H., Jr., *Dynamics of Physical Systems.* McGraw-Hill, New York, 1967.

Carlson, L. E., and Griggs, G. E. *Aluminum Catenary System Quarterly Report*. Technical Report Contract Number DOT-FR-9154, U.S. Department of Transportation, 1980.

Chignola, R., and Foroni, R. I. Estimating the Growth Kinetics of Experimental Tumors from as Few as Two Determinations of Tumor Size: Implications for Clinical Oncology. *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 5, May 2005, pp. 808–815.

Cochin, I. *Analysis and Design of Dynamic Systems.* Harper and Row, New York, 1980.

Cook, P. A. *Nonlinear Dynamical Systems.* Prentice Hall, United Kingdom, 1986.

Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

Davis, S. A., and Ledgerwood, B. K. *Electromechanical Components for Servomechanisms.* McGraw-Hill, New York, 1961.

Doebelin, E. O. *Measurement Systems Application and Design*. McGraw-Hill, New York, 1983.

Dorf, R. *Introduction to Electric Circuits*, 2d ed. Wiley, New York, 1993.

D'Souza, A. *Design of Control Systems.* Prentice Hall, Upper Saddle River, NJ, 1988.

Edelstein-Keshet, L. *Mathematical Models in Biology.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.

Elkins, J. A. *A Method for Predicting the Dynamic Response of a Pantograph Running at Constant Speed under a Finite Length of Overhead Equipment.* Technical Report TN DA36, British Railways, 1976.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems.* Addison-Wesley, Reading, MA, 1986.

Galvão, R. K. H., Yoneyama, T., and de Araújo, F. M. U. A Simple Technique for Identifying a Linearized Model for a Didactic Magnetic Levitation System. *IEEE Transactions on Education*, vol. 46, no. 1, February 2003, pp. 22–25.

Hsu, J. C., and Meyer, A. U. *Modern Control Principles and Applications.* McGraw-Hill, New York, 1968.

Johansson, R., Magnusson, M., and Åkesson, M. Identification of Human Postural Dynamics. *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 10, October 1988, pp. 858–869.

Kailath, T. *Linear Systems.* Prentice Hall, Upper Saddle River, NJ, 1980.

Kermurjian, A. From the Moon Rover to the Mars Rover. *The Planetary Report*, July/August 1990, pp. 4–11.

Krieg, M., and Mohseni, K. Developing a Transient Model for Squid Inspired Thrusters, and Incorporation into Underwater Robot Control Design. *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, France*, September 2008.

Kuo, F. F. *Network Analysis and Synthesis.* Wiley, New York, 1966.

Lago, G., and Benningfield, L. M. *Control System Theory.* Ronald Press, New York, 1962.

Lin, J.-S., and Kanellakopoulos, I. Nonlinear Design of Active Suspensions. *IEEE Control Systems Magazine*, vol. 17, issue 3, June 1997, pp. 45–59.

Mablekos, V E. *Electric Machine Theory for Power Engineers.* Harper & Row, Cambridge, MA, 1980.

Marttinen, A., Virkkunen, J., and Salminen, R. T. Control Study with Pilot Crane. *IEEE Transactions on Education*, vol. 33, no. 3, August 1990, pp. 298–305.

Milsum, J. H. *Biological Control Systems Analysis.* McGraw-Hill, New York, 1966.

Minorsky, N. *Theory of Nonlinear Control Systems.* McGraw-Hill, New York, 1969.

Nilsson, J. W., and Riedel, S. A. *Electric Circuits*, 5th ed. Addison-Wesley, Reading, MA, 1996.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. l–4.

Ogata, K. *Modern Control Engineering*, 2d ed. Prentice Hall, Upper Saddle River, NJ, 1990.

Raven, F. H. *Automatic Control Engineering*, 5th ed. McGraw-Hill, New York, 1995.

Schnell, S., and Mendoza, C. The Condition for Pseudo-First-Order Kinetics in Enzymatic Reactions Is Independent of the Initial Enzyme Concentration. *Biophysical Chemistry* (107), 2004, pp. 165–174.

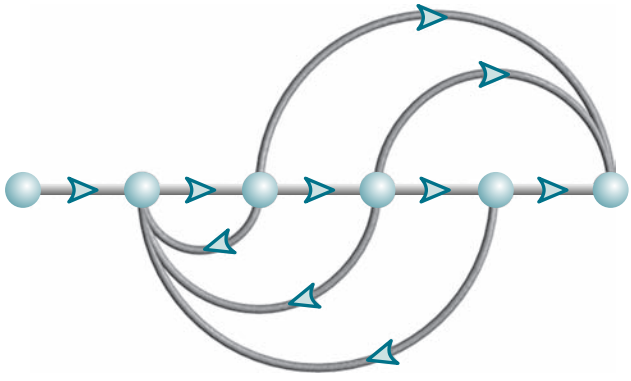
Van Valkenburg, M. E. *Network Analysis.* Prentice Hall, Upper Saddle River, NJ, 1974.

Vidyasagar, M. *Nonlinear Systems Analysis.* Prentice Hall, Upper Saddle River, NJ, 1978.

Wang, J. Z., Tie, B., Welkowitz, W., Semmlow, J. L., and Kostis, J. B. Modeling Sound Generation in Stenosed Coronary Arteries. *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 11, November 1990, pp. 1087–1094.

Modeling in the Time Domain

3



This chapter covers only state-space methods.

State Space
SS

Chapter Learning Outcomes

After completing this chapter, the student will be able to:

- Find a mathematical model, called a *state-space* representation, for a linear, time-invariant system (Sections 3.1–3.3)
- Model electrical and mechanical systems in state space (Section 3.4)
- Convert a transfer function to state space (Section 3.5)
- Convert a state-space representation to a transfer function (Section 3.6)
- Linearize a state-space representation (Section 3.7)

Case Study Learning Outcomes

You will be able to demonstrate your knowledge of the chapter objectives with case studies as follows:

- Given the antenna azimuth position control system shown on the front endpapers, you will be able to find the state-space representation of each subsystem.
- Given a description of the way a pharmaceutical drug flows through a human being, you will be able to find the state-space representation to determine drug concentrations in specified compartmentalized blocks of the process and of the human body. You will also be able to apply the same concepts to an aquifer to find water level.

3.1 Introduction

Two approaches are available for the analysis and design of feedback control systems. The first, which we began to study in Chapter 2, is known as the *classical*, or *frequency-domain*, technique. This approach is based on converting a system's differential equation to a transfer function, thus generating a mathematical model of the system that *algebraically* relates a representation of the output to a representation of the input. Replacing a differential equation with an algebraic equation not only simplifies the representation of individual subsystems but also simplifies modeling interconnected subsystems.

The primary disadvantage of the classical approach is its limited applicability: It can be applied only to linear, time-invariant systems or systems that can be approximated as such.

A major advantage of frequency-domain techniques is that they rapidly provide stability and transient response information. Thus, we can immediately see the effects of varying system parameters until an acceptable design is met.

With the arrival of space exploration, requirements for control systems increased in scope. Modeling systems by using linear, time-invariant differential equations and subsequent transfer functions became inadequate. The *state-space* approach (also referred to as the *modern*, or *time-domain*, approach) is a unified method for modeling, analyzing, and designing a wide range of systems. For example, the state-space approach can be used to represent nonlinear systems that have backlash, saturation, and dead zone. Also, it can handle, conveniently, systems with nonzero initial conditions. Time-varying systems, (for example, missiles with varying fuel levels or lift in an aircraft flying through a wide range of altitudes) can be represented in state space. Many systems do not have just a single input and a single output. Multiple-input, multiple-output systems (such as a vehicle with input direction and input velocity yielding an output direction and an output velocity) can be compactly represented in state space with a model similar in form and complexity to that used for single-input, single-output systems. The time-domain approach can be used to represent systems with a digital computer in the loop or to model systems for digital simulation. With a simulated system, system response can be obtained for changes in system parameters—an important design tool. The state-space approach is also attractive because of the availability of numerous state-space software packages for the personal computer.

The time-domain approach can also be used for the same class of systems modeled by the classical approach. This alternate model gives the control systems designer another perspective from which to create a design. While the state-space approach can be applied to a wide range of systems, it is not as intuitive as the classical approach. The designer has to engage in several calculations before the physical interpretation of the model is apparent, whereas in classical control a few quick calculations or a graphic presentation of data rapidly yields the physical interpretation.

In this book, the coverage of state-space techniques is to be regarded as an introduction to the subject, a springboard to advanced studies, and an alternate approach to frequency-domain techniques. We will limit the state-space approach to linear, time-invariant systems or systems that can be linearized by the methods of Chapter 2. The study of other classes of systems is beyond the scope of this book. Since state-space analysis and design rely on matrices and matrix operations, you may want to review this topic in Appendix G, located at www.wiley.com/college/nise, before continuing.

3.2 Some Observations

We proceed now to establish the state-space approach as an alternate method for representing physical systems. This section sets the stage for the formal definition of the state-space representation by making some observations about systems and their variables. In the discussion that follows, some of the development has been placed in footnotes to avoid clouding the main issues with an excess of equations and to ensure that the concept is clear. Although we use two electrical networks to illustrate the concepts, we could just as easily have used a mechanical or any other physical system.

We now demonstrate that for a system with many variables, such as inductor voltage, resistor voltage, and capacitor charge, we need to use differential equations only to solve for a selected subset of system variables because all other remaining system variables can be evaluated algebraically from the variables in the subset. Our examples take the following approach:

1. We select a particular *subset* of all possible system variables and call the variables in this subset *state variables.*
2. For an nth-order system, we write n *simultaneous, first-order differential equations* in terms of the state variables. We call this system of simultaneous differential equations *state equations.*
3. If we know the initial condition of all of the state variables at t_0 as well as the system input for $t \geq t_0$, we can solve the simultaneous differential equations for the state variables for $t \geq t_0$.
4. We *algebraically* combine the state variables with the system's input and find all of the other system variables for $t \geq t_0$. We call this algebraic equation the *output equation*.
5. We consider the state equations and the output equations a viable representation of the system. We call this representation of the system a *state-space representation*.

Let us now follow these steps through an example. Consider the RL network shown in Figure 3.1 with an initial current of $i(0)$.

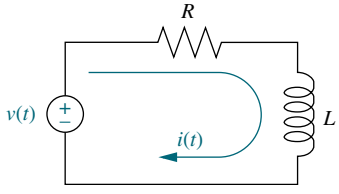


FIGURE 3.1 RL network

1. We select the current, $i(t)$, for which we will write and solve a differential equation using Laplace transforms.
2. We write the loop equation,

$$L\frac{di}{dt} + Ri = v(t) \tag{3.1}$$

3. Taking the Laplace transform, using Table 2.2, Item 7, and including the initial conditions, yields

$$L[sI(s) - i(0)] + RI(s) = V(s) \tag{3.2}$$

Assuming the input, $v(t)$, to be a unit step, $u(t)$, whose Laplace transform is $V(s) = 1/s$, we solve for $I(s)$ and get

$$I(s) = \frac{1}{R}\left(\frac{1}{s} - \frac{1}{s + \dfrac{R}{L}}\right) + \frac{i(0)}{s + \dfrac{R}{L}} \tag{3.3}$$

from which

$$i(t) = \frac{1}{R}\left(1 - e^{-(R/L)t}\right) + i(0)e^{-(R/L)t} \tag{3.4}$$

The function $i(t)$ is a subset of all possible network variables that we are able to find from Eq. (3.4) if we know its initial condition, $i(0)$, and the input, $v(t)$. Thus, $i(t)$ is a state variable, and the differential equation (3.1) is a *state equation*.

4. We can now solve for all of the other network variables *algebraically* in terms of $i(t)$ and the applied voltage, $v(t)$. For example, the voltage across the resistor is

$$v_R(t) = Ri(t) \tag{3.5}$$

The voltage across the inductor is

$$v_L(t) = v(t) - Ri(t) \tag{3.6}^{1}$$

The derivative of the current is

$$\frac{di}{dt} = \frac{1}{L}[v(t) - Ri(t)] \tag{3.7}^{2}$$

Thus, knowing the state variable, $i(t)$, and the input, $v(t)$, we can find the value, or *state*, of any network variable at any time, $t \geq t_0$. Hence, the algebraic equations, Eqs. (3.5) through (3.7), are *output equations*.

5. Since the variables of interest are completely described by Eq. (3.1) and Eqs. (3.5) through (3.7), we say that the combined state equation (3.1) and the output equations (3.5 through 3.7) form a viable representation of the network, which we call a *state-space representation*.

Equation (3.1), which describes the dynamics of the network, is not unique. This equation could be written in terms of any other network variable. For example, substituting $i = v_R/R$ into Eq. (3.1) yields

$$\frac{L}{R}\frac{dv_R}{dt} + v_R = v(t) \tag{3.8}$$

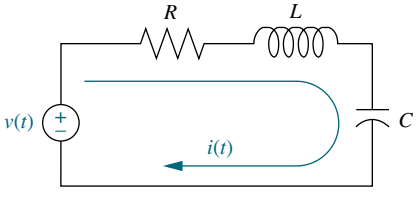


FIGURE 3.2 *RLC* network

which can be solved knowing that the initial condition $v_R(0) = Ri(0)$ and knowing $v(t)$. In this case, the state variable is $v_R(t)$. Similarly, all other network variables can now be written in terms of the state variable, $v_R(t)$, and the input, $v(t)$. Let us now extend our observations to a second-order system, such as that shown in Figure 3.2.

1. Since the network is of second order, two simultaneous, first-order differential equations are needed to solve for two state variables. We select $i(t)$ and $q(t)$, the charge on the capacitor, as the two state variables.

2. Writing the loop equation yields

$$L\frac{di}{dt} + Ri + \frac{1}{C}\int i\,dt = v(t) \tag{3.9}$$

---

[1] Since $v_L(t) = v(t) - v_R(t) = v(t) - Ri(t)$.

[2] Since $\frac{di}{dt} = \frac{1}{L}v_L(t) = \frac{1}{L}[v(t) - Ri(t)]$.

Converting to charge, using $i(t) = dq/dt$, we get

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = v(t) \tag{3.10}$$

But an nth-order differential equation can be converted to n simultaneous first-order differential equations, with each equation of the form

$$\frac{dx_i}{dt} = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + b_i f(t) \tag{3.11}$$

where each x_i is a state variable, and the a_{ij}'s and b_i are constants for linear, time-invariant systems. We say that the right-hand side of Eq. (3.11) is a *linear combination* of the state variables and the input, $f(t)$.

We can convert Eq. (3.10) into two simultaneous, first-order differential equations in terms of $i(t)$ and $q(t)$. The first equation can be $dq/dt = i$. The second equation can be formed by substituting $\int i\,dt = q$ into Eq. (3.9) and solving for di/dt. Summarizing the two resulting equations, we get

$$\frac{dq}{dt} = i \tag{3.12a}$$

$$\frac{di}{dt} = -\frac{1}{LC}q - \frac{R}{L}i + \frac{1}{L}v(t) \tag{3.12b}$$

3. These equations are the state equations and can be solved simultaneously for the state variables, $q(t)$ and $i(t)$, using the Laplace transform and the methods of Chapter 2, if we know the initial conditions for $q(t)$ and $i(t)$ and if we know $v(t)$, the input.

4. From these two state variables, we can solve for all other network variables. For example, the voltage across the inductor can be written in terms of the solved state variables and the input as

$$v_L(t) = -\frac{1}{C}q(t) - Ri(t) + v(t) \tag{3.13}$$[3]

Equation (3.13) is an *output equation*; we say that $v_L(t)$ is a *linear combination* of the state variables, $q(t)$ and $i(t)$, and the input, $v(t)$.

5. The combined state equations (3.12) and the output equation (3.13) form a viable representation of the network, which we call a *state-space representation*.

Another choice of two state variables can be made, for example, $v_R(t)$ and $v_C(t)$, the resistor and capacitor voltage, respectively. The resulting set of simultaneous, first-order differential equations follows:

$$\frac{dv_R}{dt} = -\frac{R}{L}v_R - \frac{R}{L}v_C + \frac{R}{L}v(t) \tag{3.14a}$$[4]

$$\frac{dv_C}{dt} = \frac{1}{RC}v_R \tag{3.14b}$$

[3] Since $v_L(t) = L(di/dt) = -(1/C)q - Ri + v(t)$, where di/dt can be found from Eq. (3.9), and $\int i\,dt = q$.

[4] Since $v_R(t) = i(t)R$, and $v_C(t) = (1/C)\int i\,dt$, differentiating $v_R(t)$ yields $dv_R/dt = R(di/dt) = (R/L)v_L = (R/L)[v(t) - v_R - v_C]$, and differentiating $v_C(t)$ yields $dv_C/dt = (1/C)i = (1/RC)v_R$.

Again, these differential equations can be solved for the state variables if we know the initial conditions along with $v(t)$. Further, all other network variables can be found as a linear combination of these state variables.

Is there a restriction on the choice of state variables? Yes! Typically, the minimum number of state variables required to describe a system equals the order of the differential equation. Thus, a second-order system requires a minimum of two state variables to describe it. We can define more state variables than the minimal set; however, within this minimal set the state variables must be linearly independent. For example, if $v_R(t)$ is chosen as a state variable, then $i(t)$ cannot be chosen, because $v_R(t)$ can be written as a linear combination of $i(t)$, namely $v_R(t) = Ri(t)$. Under these circumstances we say that the state variables are *linearly dependent*. State variables must be *linearly independent*; that is, no state variable can be written as a linear combination of the other state variables, or else we would not have enough information to solve for all other system variables, and we could even have trouble writing the simultaneous equations themselves.

The state and output equations can be written in vector-matrix form if the system is linear. Thus, Eq. (3.12), the state equations, can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{3.15}$$

where

$$\dot{\mathbf{x}} = \begin{bmatrix} dq/dt \\ di/dt \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1/LC & -R/L \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} q \\ i \end{bmatrix}; \qquad \mathbf{B} = \begin{bmatrix} 0 \\ 1/L \end{bmatrix}; \; u = v(t)$$

Equation (3.13), the output equation, can be written as

$$y = \mathbf{C}\mathbf{x} + Du \tag{3.16}$$

where

$$y = v_L(t); \quad \mathbf{C} = [-1/C \quad -R]; \quad \mathbf{x} = \begin{bmatrix} q \\ i \end{bmatrix}; \quad D = 1; \quad u = v(t)$$

We call the combination of Eqs. (3.15) and (3.16) a *state-space representation* of the network of Figure 3.2. A state-space representation, therefore, consists of (1) the simultaneous, first-order differential equations from which the state variables can be solved and (2) the algebraic output equation from which all other system variables can be found. A state-space representation is not unique, since a different choice of state variables leads to a different representation of the same system.

In this section, we used two electrical networks to demonstrate some principles that are the foundation of the state-space representation. The representations developed in this section were for single-input, single-output systems, where y,D, and u in Eqs. (3.15) and (3.16) are scalar quantities. In general, systems have multiple inputs and multiple outputs. For these cases, y and u become vector quantities, and D becomes a matrix. In Section 3.3 we will generalize the representation for multiple-input, multiple-output systems and summarize the concept of the state-space representation.

3.3 The General State-Space Representation

Now that we have represented a physical network in state space and have a good idea of the terminology and the concept, let us summarize and generalize the representation for linear differential equations. First, we formalize some of the definitions that we came across in the last section.

Linear combination. A linear combination of n variables, x_i, for $i = 1$ to n, is given by the following sum, S:

$$S = K_n x_n + K_{n-1} x_{n-1} + \cdots + K_1 x_1 \tag{3.17}$$

where each K_i is a constant.

Linear independence. A set of variables is said to be linearly independent if none of the variables can be written as a linear combination of the others. For example, given x_1, x_2, and x_3, if $x_2 = 5x_1 + 6x_3$, then the variables are not linearly independent, since one of them can be written as a linear combination of the other two. Now, what must be true so that one variable cannot be written as a linear combination of the other variables? Consider the example $K_2 x_2 = K_1 x_1 + K_3 x_3$. If no $x_i = 0$, then any x_i can be written as a linear combination of other variables, unless all $K_i = 0$. Formally, then, variables x_i, for $i = 1$ to n, are said to be linearly independent if their linear combination, S, equals zero *only* if every $K_i = 0$ and *no* $x_i = 0$ for all $t \geq 0$.

System variable. Any variable that responds to an input or initial conditions in a system.

State variables. The smallest set of linearly independent system variables such that the values of the members of the set at time t_0 along with known forcing functions completely determine the value of all system variables for all $t \geq t_0$.

State vector. A vector whose elements are the state variables.

State space. The n-dimensional space whose axes are the state variables. This is a new term and is illustrated in Figure 3.3, where the state variables are assumed to be a resistor voltage, v_R, and a capacitor voltage, v_C. These variables form the axes of the *state space.* A trajectory can be thought of as being mapped out by the state vector, $\mathbf{x}(t)$, for a range of t. Also shown is the state vector at the particular time $t = 4$.

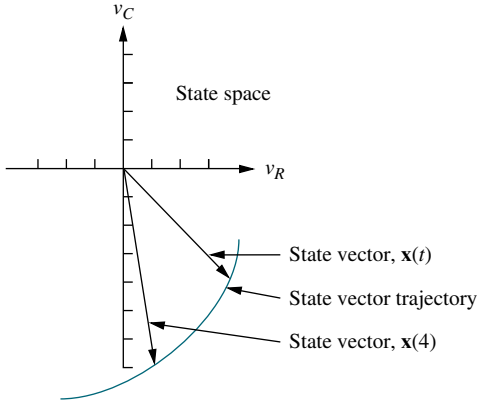


FIGURE 3.3 Graphic representation of state space and a state vector

State equations. A set of n simultaneous, first-order differential equations with n variables, where the n variables to be solved are the state variables.

Output equation. The algebraic equation that expresses the output variables of a system as linear combinations of the state variables and the inputs.

Now that the definitions have been formally stated, we define the state-space representation of a system. A system is represented in state space by the following equations:

$$\boxed{\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}} \tag{3.18}$$

$$\boxed{\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}} \tag{3.19}$$

for $t \geq t_0$ and initial conditions, $\mathbf{x}(t_0)$, where

$\mathbf{x}$ = state vector
$\dot{\mathbf{x}}$ = derivative of the state vector with respect to time
$\mathbf{y}$ = output vector
$\mathbf{u}$ = input or control vector
$\mathbf{A}$ = system matrix
$\mathbf{B}$ = input matrix
$\mathbf{C}$ = output matrix
$\mathbf{D}$ = feedforward matrix

Equation (3.18) is called the *state equation*, and the vector $\mathbf{x}$, the *state vector*, contains the state variables. Equation (3.18) can be solved for the state variables, which we demonstrate in Chapter 4. Equation (3.19) is called the *output equation*. This equation is used to calculate any other system variables. This representation of a system provides complete knowledge of all variables of the system at any $t \geq t_0$.

As an example, for a linear, time-invariant, second-order system with a single input $v(t)$, the state equations could take on the following form:

$$\frac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + b_1v(t) \tag{3.20a}$$

$$\frac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2 + b_2v(t) \tag{3.20b}$$

where x_1 and x_2 are the state variables. If there is a single output, the output equation could take on the following form:

$$y = c_1x_1 + c_2x_2 + d_1v(t) \tag{3.21}$$

The choice of state variables for a given system is not unique. The requirement in choosing the state variables is that they be linearly independent and that a minimum number of them be chosen.

3.4 Applying the State-Space Representation

In this section, we apply the state-space formulation to the representation of more complicated physical systems. The first step in representing a system is to select the state vector, which must be chosen according to the following considerations:

1. A minimum number of state variables must be selected as components of the state vector. This minimum number of state variables is sufficient to describe completely the state of the system.
2. The components of the state vector (that is, this minimum number of state variables) must be linearly independent.

Let us review and clarify these statements.

Linearly Independent State Variables

The components of the state vector must be linearly independent. For example, following the definition of linear independence in Section 3.3, if x_1, x_2, and x_3 are chosen as state variables, but $x_3 = 5x_1 + 4x_2$, then x_3 is not linearly independent of x_1

and x_2, since knowledge of the values of x_1 and x_2 will yield the value of x_3. Variables and their successive derivatives are linearly independent. For example, the voltage across an inductor, v_L, is linearly independent of the current through the inductor, i_L, since $v_L = L di_L/dt$. Thus, v_L cannot be evaluated as a linear combination of the current, i_L.

Minimum Number of State Variables

How do we know the minimum number of state variables to select? Typically, the minimum number required equals the order of the differential equation describing the system. For example, if a third-order differential equation describes the system, then three simultaneous, first-order differential equations are required along with three state variables. From the perspective of the transfer function, the order of the differential equation is the order of the denominator of the transfer function after canceling common factors in the numerator and denominator.

In most cases, another way to determine the number of state variables is to count the number of independent energy-storage elements in the system.[5] The number of these energy-storage elements equals the order of the differential equation and the number of state variables. In Figure 3.2 there are two energy-storage elements, the capacitor and the inductor. Hence, two state variables and two state equations are required for the system.

If too few state variables are selected, it may be impossible to write particular output equations, since some system variables cannot be written as a linear combination of the reduced number of state variables. In many cases, it may be impossible even to complete the writing of the state equations, since the derivatives of the state variables cannot be expressed as linear combinations of the reduced number of state variables.

If you select the minimum number of state variables but they are not linearly independent, at best you may not be able to solve for all other system variables. At worst you may not be able to complete the writing of the state equations.

Often the state vector includes more than the minimum number of state variables required. Two possible cases exist. Often state variables are chosen to be physical variables of a system, such as position and velocity in a mechanical system. Cases arise where these variables, although linearly independent, are also *decoupled*. That is, some linearly independent variables are not required in order to solve for any of the other linearly independent variables or any other dependent system variable. Consider the case of a mass and viscous damper whose differential equation is $M\,dv/dt + Dv = f(t)$, where v is the velocity of the mass. Since this is a first-order equation, one state equation is all that is required to define this system in state space with velocity as the state variable. Also, since there is only one energy-storage element, mass, only one state variable is required to represent this system in state space. However, the mass also has an associated position, which is linearly independent of velocity. If we want to include position in the state vector along with velocity, then we add position as a state variable that is linearly independent of the other state variable, velocity. Figure 3.4 illustrates what is happening. The first block is the transfer

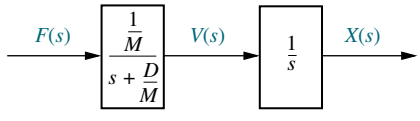


FIGURE 3.4 Block diagram of a mass and damper

[5] Sometimes it is not apparent in a schematic how many independent energy-storage elements there are. It is possible that more than the minimum number of energy-storage elements could be selected, leading to a state vector whose components number more than the minimum required and are not linearly independent. Selecting additional dependent energy-storage elements results in a system matrix of higher order and more complexity than required for the solution of the state equations.

function equivalent to $M\,dv(t)/dt + Dv(t) = f(t)$. The second block shows that we integrate the output velocity to yield output displacement (see Table 2.2, Item 10). Thus, if we want displacement as an output, the denominator, or characteristic equation, has increased in order to 2, the product of the two transfer functions. Many times, the writing of the state equations is simplified by including additional state variables.

Another case that increases the size of the state vector arises when the added variable is not linearly independent of the other members of the state vector. This usually occurs when a variable is selected as a state variable but its dependence on the other state variables is not immediately apparent. For example, energy-storage elements may be used to select the state variables, and the dependence of the variable associated with one energy-storage element on the variables of other energy-storage elements may not be recognized. Thus, the dimension of the system matrix is increased unnecessarily, and the solution for the state vector, which we cover in Chapter 4, is more difficult. Also, adding dependent state variables affects the designer's ability to use state-space methods for design.[6]

We saw in Section 3.2 that the state-space representation is not unique. The following example demonstrates one technique for selecting state variables and representing a system in state space. Our approach is to write the simple derivative equation for each energy-storage element and solve for each derivative term as a linear combination of any of the system variables and the input that are present in the equation. Next we select each differentiated variable as a state variable. Then we express all other system variables in the equations in terms of the state variables and the input. Finally, we write the output variables as linear combinations of the state variables and the input.

Example 3.1

Representing an Electrical Network

PROBLEM: Given the electrical network of Figure 3.5, find a state-space representation if the output is the current through the resistor.

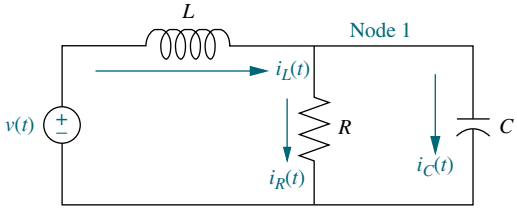


FIGURE 3.5 Electrical network for representation in state space

SOLUTION: The following steps will yield a viable representation of the network in state space.

Step 1 Label all of the branch currents in the network. These include i_L, i_R, and i_C, as shown in Figure 3.5.

[6] See Chapter 12 for state-space design techniques.

Step 2 Select the state variables by writing the derivative equation for all energy-storage elements, that is, the inductor and the capacitor. Thus,

$$C\frac{dv_C}{dt} = i_C \tag{3.22}$$

$$L\frac{di_L}{dt} = v_L \tag{3.23}$$

From Eqs. (3.22) and (3.23), choose the state variables as the quantities that are differentiated, namely v_C and i_L. Using Eq. (3.20) as a guide, we see that the state-space representation is complete if the right-hand sides of Eqs. (3.22) and (3.23) can be written as linear combinations of the state variables and the input.

Since i_C and v_L are not state variables, our next step is to express i_C and v_L as linear combinations of the state variables, v_C and i_L, and the input, $v(t)$.

Step 3 Apply network theory, such as Kirchhoff's voltage and current laws, to obtain i_C and v_L in terms of the state variables, v_C and i_L. At Node 1,

$$\begin{aligned} i_C &= -i_R + i_L \\ &= -\frac{1}{R}v_C + i_L \end{aligned} \tag{3.24}$$

which yields i_C in terms of the state variables, v_C and i_L.

Around the outer loop,

$$v_L = -v_C + v(t) \tag{3.25}$$

which yields v_L in terms of the state variable, v_C, and the source, $v(t)$.

Step 4 Substitute the results of Eqs. (3.24) and (3.25) into Eqs. (3.22) and (3.23) to obtain the following state equations:

$$C\frac{dv_C}{dt} = -\frac{1}{R}v_C + i_L \tag{3.26a}$$

$$L\frac{di_L}{dt} = \quad -v_C + v(t) \tag{3.26b}$$

or

$$\frac{dv_C}{dt} = -\frac{1}{RC}v_C + \frac{1}{C}i_L \tag{3.27a}$$

$$\frac{di_L}{dt} = \quad -\frac{1}{L}v_C + \frac{1}{L}v(t) \tag{3.27b}$$

Step 5 Find the output equation. Since the output is $i_R(t)$,

$$i_R = \frac{1}{R}v_C \tag{3.28}$$

The final result for the state-space representation is found by representing Eqs. (3.27) and (3.28) in vector-matrix form as follows:

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -1/(RC) & 1/C \\ -1/L & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} v(t) \tag{3.29a}$$

$$i_R = \begin{bmatrix} 1/R & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} \tag{3.29b}$$

where the dot indicates differentiation with respect to time.

In order to clarify the representation of physical systems in state space, we will look at two more examples. The first is an electrical network with a dependent source. Although we will follow the same procedure as in the previous problem, this problem will yield increased complexity in applying network analysis to find the state equations. For the second example, we find the state-space representation of a mechanical system.

Example 3.2

Representing an Electrical Network with a Dependent Source

PROBLEM: Find the state and output equations for the electrical network shown in Figure 3.6 if the output vector is $\mathbf{y} = [\,v_{R_2} \quad i_{R_2}\,]^T$, where T means transpose.[7]

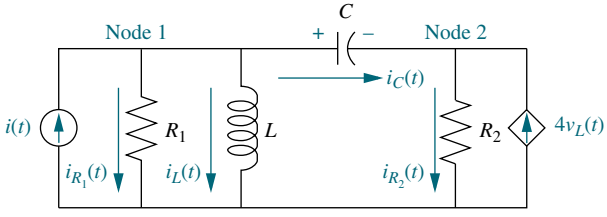


FIGURE 3.6 Electrical network for Example 3.2

SOLUTION: Immediately notice that this network has a voltage-dependent current source.

Step 1 Label all of the branch currents on the network, as shown in Figure 3.6.

Step 2 Select the state variables by listing the voltage-current relationships for all of the energy-storage elements:

$$L\frac{di_L}{dt} = v_L \tag{3.30a}$$

$$C\frac{dv_C}{dt} = i_C \tag{3.30b}$$

From Eqs. (3.30) select the state variables to be the differentiated variables. Thus, the state variables, x_1 and x_2, are

$$x_1 = i_L; \quad x_2 = v_C \tag{3.31}$$

Step 3 Remembering that the form of the state equation is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.32}$$

we see that the remaining task is to transform the right-hand side of Eq. (3.30) into linear combinations of the state variables and input source current. Using Kirchhoff's voltage and current laws, we find v_L and i_C in terms of the state variables and the input current source.

[7] See Appendix G for a discussion of the transpose. Appendix G is located at www.wiley.com/college/nise.

Around the mesh containing L and C,

$$v_L = v_C + v_{R_2} = v_C + i_{R_2}R_2 \tag{3.33}$$

But at Node 2, $i_{R_2} = i_C + 4v_L$. Substituting this relationship for i_{R_2} into Eq. (3.33) yields

$$v_L = v_C + (i_C + 4v_L)R_2 \tag{3.34}$$

Solving for v_L, we get

$$v_L = \frac{1}{1 - 4R_2}(v_C + i_C R_2) \tag{3.35}$$

Notice that since v_C is a state variable, we only need to find i_C in terms of the state variables. We will then have obtained v_L in terms of the state variables.

Thus, at Node 1 we can write the sum of the currents as

$$\begin{aligned} i_C &= i(t) - i_{R_1} - i_L \\ &= i(t) - \frac{v_{R_1}}{R_1} - i_L \\ &= i(t) - \frac{v_L}{R_1} - i_L \end{aligned} \tag{3.36}$$

where $v_{R_1} = v_L$. Equations (3.35) and (3.36) are two equations relating v_L and i_C in terms of the state variables i_L and v_C. Rewriting Eqs. (3.35) and (3.36), we obtain two simultaneous equations yielding v_L and i_C as linear combinations of the state variables i_L and v_C:

$$(1 - 4R_2)v_L - R_2 i_C = v_C \tag{3.37a}$$

$$-\frac{1}{R_1}v_L - i_C = i_L - i(t) \tag{3.37b}$$

Solving Eq. (3.37) simultaneously for v_L and i_C yields

$$v_L = \frac{1}{\Delta}[R_2 i_L - v_C - R_2 i(t)] \tag{3.38}$$

and

$$i_C = \frac{1}{\Delta}\left[(1 - 4R_2)i_L + \frac{1}{R_1}v_C - (1 - 4R_2)i(t)\right] \tag{3.39}$$

where

$$\Delta = -\left[(1 - 4R_2) + \frac{R_2}{R_1}\right] \tag{3.40}$$

Substituting Eqs. (3.38) and (3.39) into (3.30), simplifying, and writing the result in vector-matrix form renders the following state equation:

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} R_2/(L\Delta) & -1/(L\Delta) \\ (1 - 4R_2)/(C\Delta) & 1/(R_1 C\Delta) \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} -R_2/(L\Delta) \\ -(1 - 4R_2)/(C\Delta) \end{bmatrix} i(t) \tag{3.41}$$

Step 4 Derive the output equation. Since the specified output variables are v_{R_2} and i_{R_2}, we note that around the mesh containing C, L, and R_2,

$$v_{R_2} = -v_C + v_L \tag{3.42a}$$

$$i_{R_2} = i_C + 4v_L \tag{3.42b}$$

Substituting Eqs. (3.38) and (3.39) into Eq. (3.42), v_{R_2} and i_{R_2} are obtained as linear combinations of the state variables, i_L and v_C. In vector-matrix form, the output equation is

$$\begin{bmatrix} v_{R_2} \\ i_{R_2} \end{bmatrix} = \begin{bmatrix} R_2/\Delta & -(1+1/\Delta) \\ 1/\Delta & (1-4R_1)/(\Delta R_1) \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} -R_2/\Delta \\ -1/\Delta \end{bmatrix} i(t) \tag{3.43}$$

In the next example, we find the state-space representation for a mechanical system. It is more convenient when working with mechanical systems to obtain the state equations directly from the equations of motion rather than from the energy-storage elements. For example, consider an energy-storage element such as a spring, where $F = Kx$. This relationship does not contain the derivative of a physical variable as in the case of electrical networks, where $i = C\,dv/dt$ for capacitors, and $v = L\,di/dt$ for inductors. Thus, in mechanical systems we change our selection of state variables to be the position and velocity of each point of linearly independent motion. In the example, we will see that although there are three energy-storage elements, there will be four state variables; an additional linearly independent state variable is included for the convenience of writing the state equations. It is left to the student to show that this system yields a fourth-order transfer function if we relate the displacement of either mass to the applied force, and a third-order transfer function if we relate the velocity of either mass to the applied force.

Example 3.3

Representing a Translational Mechanical System

PROBLEM: Find the state equations for the translational mechanical system shown in Figure 3.7.

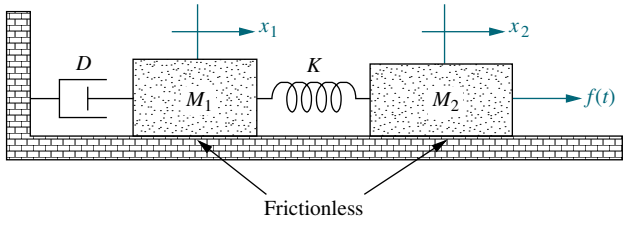


FIGURE 3.7 Translational mechanical system

SOLUTION: First write the differential equations for the network in Figure 3.7, using the methods of Chapter 2 to find the Laplace-transformed equations of motion. Next take the inverse Laplace transform of these equations, assuming zero

initial conditions, and obtain

$$M_1\frac{d^2x_1}{dt^2}+D\frac{dx_1}{dt}+Kx_1-Kx_2=0 \tag{3.44}$$

$$-Kx_1+M_2\frac{d^2x_2}{dt^2}+Kx_2=f(t) \tag{3.45}$$

Now let $d^2x_1/dt^2 = dv_1/dt$, and $d^2x_2/dt^2 = dv_2/dt$, and then select x_1, v_1, x_2, and v_2 as state variables. Next form two of the state equations by solving Eq. (3.44) for dv_1/dt and Eq. (3.45) for dv_2/dt. Finally, add $dx_1/dt = v_1$ and $dx_2/dt = v_2$ to complete the set of state equations. Hence,

$$\frac{dx_1}{dt}= +v_1 \tag{3.46a}$$

$$\frac{dv_1}{dt}=-\frac{K}{M_1}x_1-\frac{D}{M_1}v_1+\frac{K}{M_1}x_2 \tag{3.46b}$$

$$\frac{dx_2}{dt}= +v_2 \tag{3.46c}$$

$$\frac{dv_2}{dt}=+\frac{K}{M_2}x_1-\frac{K}{M_2}x_2+\frac{1}{M_2}f(t) \tag{3.46d}$$

In vector-matrix form,

$$\begin{bmatrix}\dot{x}_1\\ \dot{v}_1\\ \dot{x}_2\\ \dot{v}_2\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ -K/M_1 & -D/M_1 & K/M_1 & 0\\ 0 & 0 & 0 & 1\\ K/M_2 & 0 & -K/M_2 & 0\end{bmatrix}\begin{bmatrix}x_1\\ v_1\\ x_2\\ v_2\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 0\\ 1/M_2\end{bmatrix}f(t) \tag{3.47}$$

where the dot indicates differentiation with respect to time. What is the output equation if the output is $x(t)$?

Skill-Assessment Exercise 3.1

PROBLEM: Find the state-space representation of the electrical network shown in Figure 3.8. The output is $v_o(t)$.

WileyPLUS
WPCS
Control Solutions

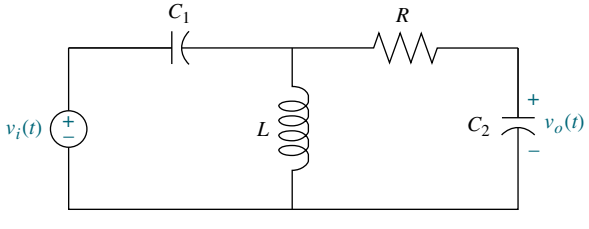


FIGURE 3.8 Electric circuit for Skill-Assessment Exercise 3.1

ANSWER:

$$\dot{\mathbf{x}} = \begin{bmatrix} 1/C_1 & 1/C_1 & -1/C_1 \\ -1/L & 0 & 0 \\ 1/C_2 & 0 & -1/C_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} v_i(t)$$

$$y = [0 \quad 0 \quad 1]\mathbf{x}$$

The complete solution is at www.wiley.com/college/nise.

Skill-Assessment Exercise 3.2

PROBLEM: Represent the translational mechanical system shown in Figure 3.9 in state space, where $x_3(t)$ is the output.

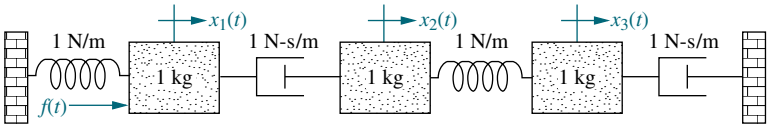


FIGURE 3.9 Translational mechanical system for Skill-Assessment Exercise 3.2

ANSWER:

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} f(t)$$

$$y = [\,0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0\,]\mathbf{z}$$

where

$$\mathbf{z} = [\,x_1 \quad \dot{x}_1 \quad x_2 \quad \dot{x}_2 \quad x_3 \quad \dot{x}_3\,]^T$$

The complete solution is at www.wiley.com/college/nise.

3.5 Converting a Transfer Function to State Space

In the last section, we applied the state-space representation to electrical and mechanical systems. We learn how to convert a transfer function representation to a state-space representation in this section. One advantage of the state-space

representation is that it can be used for the simulation of physical systems on the digital computer. Thus, if we want to simulate a system that is represented by a transfer function, we must first convert the transfer function representation to state space.

At first we select a set of state variables, called *phase variables*, where each subsequent state variable is defined to be the derivative of the previous state variable. In Chapter 5 we show how to make other choices for the state variables.

Let us begin by showing how to represent a general, nth-order, linear differential equation with constant coefficients in state space in the phase-variable form. We will then show how to apply this representation to transfer functions.

Consider the differential equation

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_0 u \tag{3.48}$$

A convenient way to choose state variables is to choose the output, $y(t)$, and its $(n-1)$ derivatives as the state variables. This choice is called the *phase-variable choice*. Choosing the state variables, x_i, we get

$$x_1 = y \tag{3.49a}$$

$$x_2 = \frac{dy}{dt} \tag{3.49b}$$

$$x_3 = \frac{d^2 y}{dt^2} \tag{3.49c}$$

$$\vdots$$

$$x_n = \frac{d^{n-1} y}{dt^{n-1}} \tag{3.49d}$$

and differentiating both sides yields

$$\dot{x}_1 = \frac{dy}{dt} \tag{3.50a}$$

$$\dot{x}_2 = \frac{d^2 y}{dt^2} \tag{3.50b}$$

$$\dot{x}_3 = \frac{d^3 y}{dt^3} \tag{3.50c}$$

$$\vdots$$

$$\dot{x}_n = \frac{d^n y}{dt^n} \tag{3.50d}$$

where the dot above the x signifies differentiation with respect to time.

Substituting the definitions of Eq. (3.49) into Eq. (3.50), the state equations are evaluated as

$$\dot{x}_1 = x_2 \tag{3.51a}$$

$$\dot{x}_2 = x_3 \tag{3.51b}$$

$$\vdots$$

$$\dot{x}_{n-1} = x_n \tag{3.51c}$$

$$\dot{x}_n = -a_0 x_1 - a_1 x_2 \cdots - a_{n-1} x_n + b_0 u \tag{3.51d}$$

where Eq. (3.51d) was obtained from Eq. (3.48) by solving for $d^n y/dt^n$ and using Eq. (3.49). In vector-matrix form, Eq. (3.51) become

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 & -a_5 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b_0 \end{bmatrix} u \tag{3.52}$$

Equation (3.52) is the phase-variable form of the state equations. This form is easily recognized by the unique pattern of 1's and 0's and the negative of the coefficients of the differential equation written in reverse order in the last row of the system matrix.

Finally, since the solution to the differential equation is $y(t)$, or x_1, the output equation is

$$y = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \tag{3.53}$$

In summary, then, to convert a transfer function into state equations in phase-variable form, we first convert the transfer function to a differential equation by cross-multiplying and taking the inverse Laplace transform, assuming zero initial conditions. Then we represent the differential equation in state space in phase-variable form. An example illustrates the process.

Example 3.4

Converting a Transfer Function with Constant Term in Numerator

PROBLEM: Find the state-space representation in phase-variable form for the transfer function shown in Figure 3.10(a).

SOLUTION:

Step 1 Find the associated differential equation. Since

$$\frac{C(s)}{R(s)} = \frac{24}{(s^3 + 9s^2 + 26s + 24)} \tag{3.54}$$

cross-multiplying yields

$$(s^3 + 9s^2 + 26s + 24)C(s) = 24R(s) \tag{3.55}$$

The corresponding differential equation is found by taking the inverse Laplace transform, assuming zero initial conditions:

$$\dddot{c} + 9\ddot{c} + 26\dot{c} + 24c = 24r \tag{3.56}$$

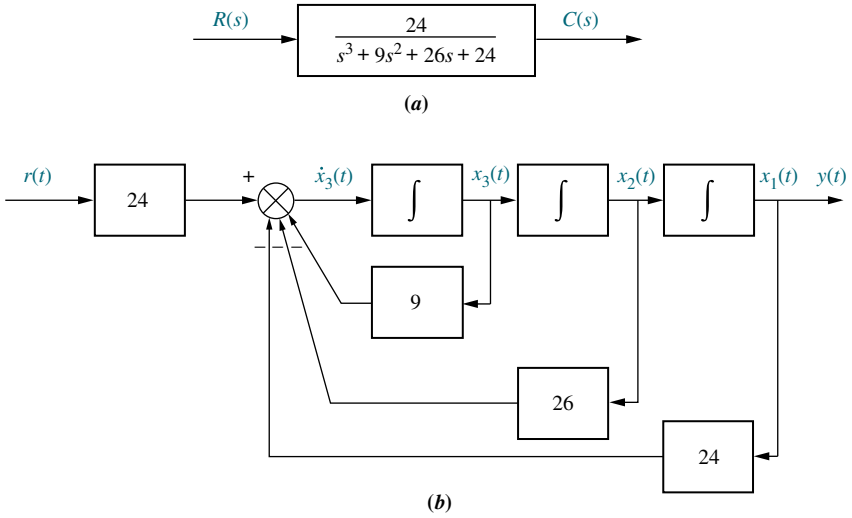


FIGURE 3.10 **a.** Transfer function; **b.** equivalent block diagram showing phase variables. Note: $y(t) = c(t)$.

Step 2 Select the state variables.
Choosing the state variables as successive derivatives, we get

$$x_1 = c \tag{3.57a}$$
$$x_2 = \dot{c} \tag{3.57b}$$
$$x_3 = \ddot{c} \tag{3.57c}$$

Differentiating both sides and making use of Eq. (3.57) to find $\dot{x}_1$ and $\dot{x}_2$, and Eq. (3.56) to find $\tilde{c} = \dot{x}_3$, we obtain the state equations. Since the output is $c = x_1$, the combined state and output equations are

$$\dot{x}_1 = x_2 \tag{3.58a}$$
$$\dot{x}_2 = x_3 \tag{3.58b}$$
$$\dot{x}_3 = -24x_1 - 26x_2 - 9x_3 + 24r \tag{3.58c}$$
$$y = c = x_1 \tag{3.58d}$$

In vector-matrix form,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -24 & -26 & -9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 24 \end{bmatrix} r \tag{3.59a}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.59b}$$

Notice that the third row of the system matrix has the same coefficients as the denominator of the transfer function but negative and in reverse order.

At this point, we can create an equivalent block diagram of the system of Figure 3.10(a) to help visualize the state variables. We draw three integral blocks as shown in Figure 3.10(b) and label each output as one of the state variables, $x_i(t)$, as shown. Since the input to each integrator is $x_i(t)$, use Eqs. (3.58a), (3.58b), and (3.58c) to determine

the combination of input signals to each integrator. Form and label each input. Finally, use Eq. (3.58d) to form and label the output, $y(t) = c(t)$. The final result of Figure 3.10 (b) is a system equivalent to Figure 3.10(a) that explicitly shows the state variables and gives a vivid picture of the state-space representation.

MATLAB ML

Students who are using MATLAB should now run ch3p1 through ch3p4 in Appendix B. You will learn how to represent the system matrix **A**, the input matrix **B**, and the output matrix **C** using MATLAB. You will learn how to convert a transfer function to the state-space representation in phase-variable form. Finally, Example 3.4 will be solved using MATLAB.

The transfer function of Example 3.4 has a constant term in the numerator. If a transfer function has a polynomial in s in the numerator that is of order less than the polynomial in the denominator, as shown in Figure 3.11(a), the numerator and denominator can be handled separately. First separate the transfer function into two cascaded transfer functions, as shown in Figure 3.11(b); the first is the denominator, and the second is just the numerator. The first transfer function with just the denominator is converted to the phase-variable representation in state space as demonstrated in the last example. Hence, phase variable x_1 is the output, and the rest of the phase variables are the internal variables of the first block, as shown in Figure 3.11(b). The second transfer function with just the numerator yields

$$Y(s) = C(s) = (b_2s^2 + b_1s + b_0)X_1(s) \tag{3.60}$$

where, after taking the inverse Laplace transform with zero initial conditions,

$$y(t) = b_2\frac{d^2x_1}{dt^2} + b_1\frac{dx_1}{dt} + b_0x_1 \tag{3.61}$$

But the derivative terms are the definitions of the phase variables obtained in the first block. Thus, writing the terms in reverse order to conform to an output equation,

$$y(t) = b_0x_1 + b_1x_2 + b_2x_3 \tag{3.62}$$

Hence, the second block simply forms a specified linear combination of the state variables developed in the first block.

From another perspective, the denominator of the transfer function yields the state equations, while the numerator yields the output equation. The next example demonstrates the process.

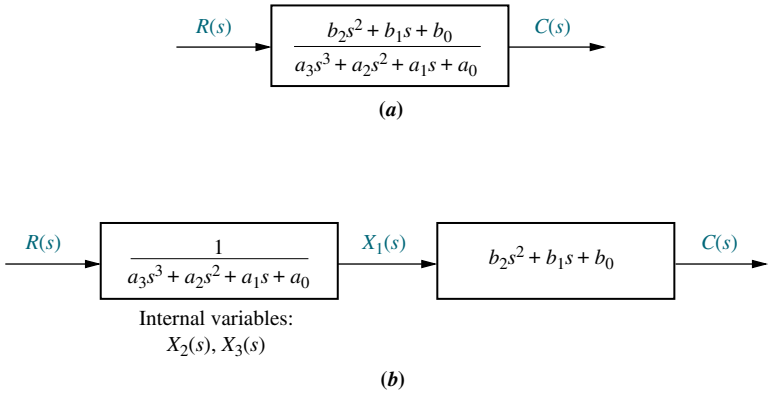


FIGURE 3.11 Decomposing a transfer function

Example 3.5

Converting a Transfer Function with Polynomial in Numerator

PROBLEM: Find the state-space representation of the transfer function shown in Figure 3.12(*a*).

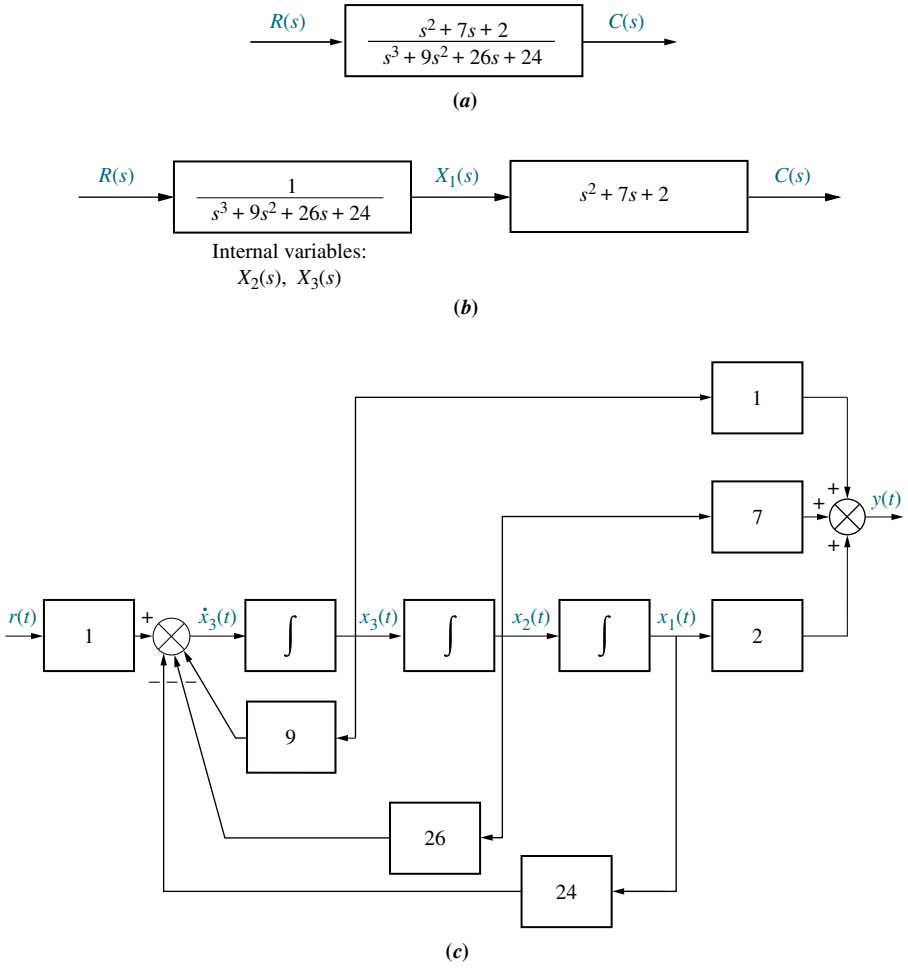


FIGURE 3.12 **a.** Transfer function; **b.** decomposed transfer function; **c.** equivalent block diagram Note: $y(t) = c(t)$.

SOLUTION: This problem differs from Example 3.4 since the numerator has a polynomial in s instead of just a constant term.

Step 1 Separate the system into two cascaded blocks, as shown in Figure 3.12(*b*). The first block contains the denominator and the second block contains the numerator.

Step 2 Find the state equations for the block containing the denominator. We notice that the first block's numerator is 1/24 that of Example 3.4. Thus, the state equations are the same except that this system's input matrix is 1/24 that of Example 3.4. Hence, the state equation is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -24 & -26 & -9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r \qquad (3.63)$$

Step 3 Introduce the effect of the block with the numerator. The second block of Figure 3.12(b), where $b_2 = 1$, $b_1 = 7$, and $b_0 = 2$, states that

$$C(s) = (b_2s^2 + b_1s + b_0)X_1(s) = (s^2 + 7s + 2)X_1(s) \tag{3.64}$$

Taking the inverse Laplace transform with zero initial conditions, we get

$$c = \ddot{x}_1 + 7\dot{x}_1 + 2x_1 \tag{3.65}$$

But

$$x_1 = x_1$$
$$\dot{x}_1 = x_2$$
$$\ddot{x}_1 = x_3$$

Hence,

$$y = c(t) = b_2x_3 + b_1x_2 + b_0x_1 = x_3 + x_2 + 2x_1 \tag{3.66}$$

Thus, the last box of Figure 3.11(b) "collects" the states and generates the output equation. From Eq. (3.66),

$$y = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 & 7 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.67}$$

Although the second block of Figure 3.12(b) shows differentiation, this block was implemented without differentiation because of the partitioning that was applied to the transfer function. The last block simply collected derivatives that were already formed by the first block.

Once again we can produce an equivalent block diagram that vividly represents our state-space model. The first block of Figure 3.12(b) is the same as Figure 3.10(a) except for the different constant in the numerator. Thus, in Figure 3.12(c) we reproduce Figure 3.10(b) except for the change in the numerator constant, which appears as a change in the input multiplying factor. The second block of Figure 3.12(b) is represented using Eq. (3.66), which forms the output from a linear combination of the state variables, as shown in Figure 3.12(c).

TryIt 3.1

Use the following MATLAB statements to form an LTI state-space representation from the transfer function shown in Figure 3.12(a). The **A** matrix and **B** vector are shown in Eq. (3.63). The **C** vector is shown in Eq. (3.67).

```
num=[1 7 2];
den=[1 9 26 24];
[A,B,C,D]=tf2ss...
 (num,den);
P=[0 0 1;0 1 0;1 0 0];
A=inv(P)*A*P
B=inv(P)*B
C=C*P
```

Skill-Assessment Exercise 3.3

WileyPLUS
WPCS
Control Solutions

PROBLEM: Find the state equations and output equation for the phase-variable representation of the transfer function $G(s) = \dfrac{2s + 1}{s^2 + 7s + 9}$.

ANSWER:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -9 & -7 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t)$$
$$y = \begin{bmatrix} 1 & 2 \end{bmatrix} \mathbf{x}$$

The complete solution is at www.wiley.com/college.nise.

3.6 Converting from State Space to a Transfer Function

In Chapters 2 and 3, we have explored two methods of representing systems: the transfer function representation and the state-space representation. In the last section, we united the two representations by converting transfer functions into state-space representations. Now we move in the opposite direction and convert the state-space representation into a transfer function.

Given the state and output equations

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.68a}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{3.68b}$$

take the Laplace transform assuming zero initial conditions:[8]

$$s\mathbf{X}(s) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \tag{3.69a}$$

$$\mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) \tag{3.69b}$$

Solving for $\mathbf{X}(s)$ in Eq. (3.69a),

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{B}\mathbf{U}(s) \tag{3.70}$$

or

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \tag{3.71}$$

where $\mathbf{I}$ is the identity matrix.

Substituting Eq. (3.71) into Eq. (3.69b) yields

$$\mathbf{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) + \mathbf{D}\mathbf{U}(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}]\mathbf{U}(s) \tag{3.72}$$

We call the matrix $[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}]$ the transfer function matrix, since it relates the output vector, $\mathbf{Y}(s)$, to the input vector, $\mathbf{U}(s)$. However, if $\mathbf{U}(s) = U(s)$ and $\mathbf{Y}(s) = Y(s)$ are scalars, we can find the transfer function,

$$\boxed{T(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{B} + \mathbf{D}} \tag{3.73}$$

Let us look at an example.

[8] The Laplace transform of a vector is found by taking the Laplace transform of each component. Since $\dot{\mathbf{x}}$ consists of the derivatives of the state variables, the Laplace transform of $\dot{\mathbf{x}}$ with zero initial conditions yields each component with the form $sX_i(s)$, where $X_i(s)$ is the Laplace transform of the state variable. Factoring out the complex variable, s, in each component yields the Laplace transform of $\dot{\mathbf{x}}$ as $s\,\mathbf{X}(s)$, where $\mathbf{X}(s)$ is a column vector with components $X_i(s)$.

Example 3.6

State-Space Representation to Transfer Function

PROBLEM: Given the system defined by Eq. (3.74), find the transfer function, $T(s) = Y(s)/U(s)$, where $U(s)$ is the input and $Y(s)$ is the output.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} u \tag{3.74a}$$

$$y = [1 \quad 0 \quad 0]\mathbf{x} \tag{3.74b}$$

SOLUTION: The solution revolves around finding the term $(s\mathbf{I} - \mathbf{A})^{-1}$ in Eq. (3.73).[9] All other terms are already defined. Hence, first find $(s\mathbf{I} - \mathbf{A})$:

$$(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} = \begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 2 & s+3 \end{bmatrix} \tag{3.75}$$

Now form $(s\mathbf{I} - \mathbf{A})^{-1}$:

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{\text{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})} = \frac{\begin{bmatrix} (s^2 + 3s + 2) & s+3 & 1 \\ -1 & s(s+3) & s \\ -s & -(2s+1) & s^2 \end{bmatrix}}{s^3 + 3s^2 + 2s + 1} \tag{3.76}$$

Substituting $(s\mathbf{I} - \mathbf{A})^{-1}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ into Eq. (3.73), where

$$\mathbf{B} = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = [1 \quad 0 \quad 0]$$

$$\mathbf{D} = 0$$

we obtain the final result for the transfer function:

$$T(s) = \frac{10(s^2 + 3s + 2)}{s^3 + 3s^2 + 2s + 1} \tag{3.77}$$

MATLAB ML

Students who are using MATLAB should now run ch3p5 in Appendix B. You will learn how to convert a state-space representation to a transfer function using MATLAB. You can practice by writing a MATLAB program to solve Example 3.6.

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch3sp1 in Appendix F located at www.wiley.com/college/nise. You will learn how to use the Symbolic Math Toolbox to write matrices and vectors. You will see that the Symbolic Math Toolbox yields an alternative way to use MATLAB to solve Example 3.6.

[9] See Appendix G. It is located at www.wiley.com/college/nise and discusses the evaluation of the matrix inverse.

Skill-Assessment Exercise 3.4

PROBLEM: Convert the state and output equations shown in Eq. (3.78) to a transfer function.

$$\dot{\mathbf{x}} = \begin{bmatrix} -4 & -1.5 \\ 4 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}u(t) \tag{3.78a}$$

$$y = [1.5 \quad 0.625]\mathbf{x} \tag{3.78b}$$

ANSWER:

$$G(s) = \frac{3s+5}{s^2+4s+6}$$

The complete solution is located at www.wiley.com/college/nise.

TryIt 3.2

Use the following MATLAB and the Control System Toolbox statements to obtain the transfer function shown in Skill-Assessment Exercise 3.4 from the state-space representation of Eq. (3.78).

```
A=[-4  -1.5;4  0];
B=[2  0]';
C=[1.5 0.625];
D=0;
T=ss(A,B,C,D);
T=tf(T)
```

In Example 3.6, the state equations in phase-variable form were converted to transfer functions. In Chapter 5, we will see that other forms besides the phase-variable form can be used to represent a system in state space. The method of finding the transfer function representation for these other forms is the same as that presented in this section.

3.7 Linearization

A prime advantage of the state-space representation over the transfer function representation is the ability to represent systems with nonlinearities, such as the one shown in Figure 3.13. The ability to represent nonlinear systems does not imply the ability to solve their state equations for the state variables and the output. Techniques do exist for the solution of some nonlinear state equations, but this study is beyond the scope of this course. However, in Appendix H, located at www.wiley.com/college/nise,

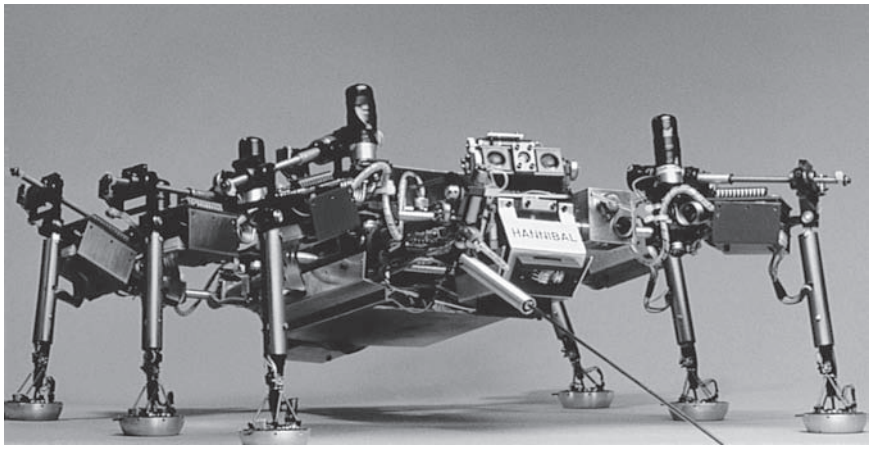


FIGURE 3.13 Walking robots, such as *Hannibal* shown here, can be used to explore hostile environments and rough terrain, such as that found on other planets or inside volcanoes.

you can see how to use the digital computer to solve state equations. This method also can be used for nonlinear state equations.

If we are interested in small perturbations about an equilibrium point, as we were when we studied linearization in Chapter 2, we can also linearize the state equations about the equilibrium point. The key to linearization about an equilibrium point is, once again, the Taylor series. In the following example, we write the state equations for a simple pendulum, showing that we can represent a nonlinear system in state space; then we linearize the pendulum about its equilibrium point, the vertical position with zero velocity.

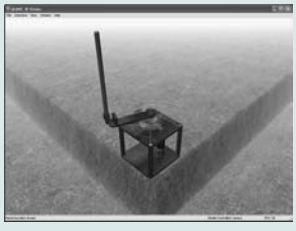
Example 3.7

Representing a Nonlinear System

PROBLEM: First represent the simple pendulum shown in Figure 3.14(a) (which could be a simple model for the leg of the robot shown in Figure 3.13) in state space: Mg is the weight, T is an applied torque in the θ direction, and L is the length of the pendulum. Assume the mass is evenly distributed, with the center of mass at $L/2$. Then linearize the state equations about the pendulum's equilibrium point—the vertical position with zero angular velocity.

Virtual Experiment 3.1 Rotary Inverted Pendulum

Put theory into practice by simulating the linear and non-linear model of the Quanser Rotary Inverted Pendulum in LabVIEW. The behavior of an inverted pendulum is similar to a variety of systems, such as Segway transporters and human posture.



Virtual experiments are found on WileyPLUS.

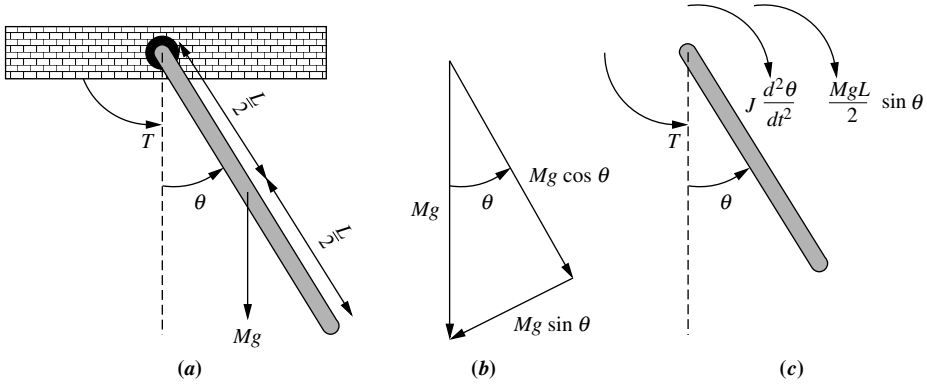


FIGURE 3.14 **a.** Simple pendulum; **b.** force components of Mg; **c.** free-body diagram

SOLUTION: First draw a free-body diagram as shown in Figure 3.14(c). Summing the torques, we get

$$J\frac{d^2\theta}{dt^2}+\frac{MgL}{2}\sin\theta = T \tag{3.79}$$

where J is the moment of inertia of the pendulum around the point of rotation. Select the state variables x_1 and x_2 as phase variables. Letting $x_1 = \theta$ and $x_2 = d\theta/dt$, we write the state equations as

$$\dot{x}_1 = x_2 \tag{3.80a}$$

$$\dot{x}_2 = -\frac{MgL}{2J}\sin x_1 + \frac{T}{J} \tag{3.80b}$$

where $\dot{x}_2 = d^2\theta/dt^2$ is evaluated from Eq. (3.79).

Thus, we have represented a nonlinear system in state space. It is interesting to note that the nonlinear Eq. (3.80) represent a valid and complete model of the pendulum in state space even under nonzero initial conditions and even if parameters are time varying. However, if we want to apply classical techniques and convert these state equations to a transfer function, we must linearize them.

Let us proceed now to linearize the equation about the equilibrium point, $x_1 = 0$, $x_2 = 0$, that is, $\theta = 0$ and $d\theta/dt = 0$. Let x_1 and x_2 be perturbed about the equilibrium point, or

$$x_1 = 0 + \delta x_1 \tag{3.81a}$$

$$x_2 = 0 + \delta x_2 \tag{3.81b}$$

Using Eq. (2.182), we obtain

$$\sin x_1 - \sin 0 = \left.\frac{d(\sin x_1)}{dx_1}\right|_{x_1=0} \delta x_1 = \delta x_1 \tag{3.82}$$

from which

$$\sin x_1 = \delta x_1 \tag{3.83}$$

Substituting Eqs. (3.81) and (3.83) into Eq. (3.80) yields the following state equations:

$$\dot{\delta} x_1 = \delta x_2 \tag{3.84a}$$

$$\dot{\delta} x_2 = -\frac{MgL}{2J}\delta x_1 + \frac{T}{J} \tag{3.84b}$$

which are linear and a good approximation to Eq. (3.80) for small excursions away from the equilibrium point. What is the output equation?

Skill-Assessment Exercise 3.5

WileyPLUS
WPCS
Control Solutions

PROBLEM: Represent the translational mechanical system shown in Figure 3.15 in state space about the equilibrium displacement. The spring is nonlinear, where the relationship between the spring force, $f_s(t)$, and the spring displacement, $x_s(t)$, is $f_s(t) = 2x_s^2(t)$. The applied force is $f(t) = 10 + \delta f(t)$, where $\delta f(t)$ is a small force about the 10 N constant value.

Assume the output to be the displacement of the mass, $x(t)$.

ANSWER:

$$\mathbf{x} = \begin{bmatrix} 0 & 1 \\ -4\sqrt{5} & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\delta f(t)$$

$$y = [1 \quad 0]\mathbf{x}$$

The complete solution is located at www.wiley.com/college/nise.

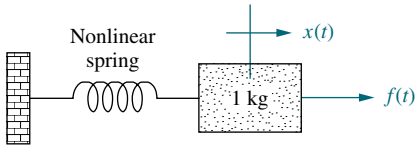


FIGURE 3.15 Nonlinear translational mechanical system for Skill-Assessment Exercise 3.5

Case Studies

Antenna Control: State-Space Representation

We have covered the state-space representation of individual physical subsystems in this chapter. In Chapter 5, we will assemble individual subsystems into feedback control systems and represent the entire feedback system in state space. Chapter 5 also shows how the state-space representation, via signal-flow diagrams, can be used to interconnect these subsystems and permit the state-space representation of the whole closed-loop system. In the following case study, we look at the antenna azimuth position control system and demonstrate the concepts of this chapter by representing each subsystem in state space.

PROBLEM: Find the state-space representation in phase-variable form for each dynamic subsystem in the antenna azimuth position control system shown on the front endpapers, *Configuration 1*. By *dynamic*, we mean that the system does not reach the steady state instantaneously. For example, a system described by a differential equation of first order or higher is a dynamic system. A pure gain, on the other hand, is an example of a nondynamic system, since the steady state is reached instantaneously.

SOLUTION: In the case study problem of Chapter 2, each subsystem of the antenna azimuth position control system was identified. We found that the power amplifier and the motor and load were dynamic systems. The preamplifier and the potentiometers are pure gains and so respond instantaneously. Hence, we will find the state-space representations only of the power amplifier and of the motor and load.

Power amplifier:

The transfer function of the power amplifier is given on the front endpapers as $G(s) = 100/(s + 100)$. We will convert this transfer function to its state-space representation. Letting $v_p(t)$ represent the power amplifier input and $e_a(t)$ represent the power amplifier output,

$$G(s) = \frac{E_a(s)}{V_p(s)} = \frac{100}{(s + 100)} \tag{3.85}$$

Cross-multiplying, $(s + 100)E_a(s) = 100V_p(s)$, from which the differential equation can be written as

$$\frac{de_a}{dt} + 100e_a = 100v_p(t) \tag{3.86}$$

Rearranging Eq. (3.86) leads to the state equation with e_a as the state variable:

$$\frac{de_a}{dt} = -100e_a + 100v_p(t) \tag{3.87}$$

Since the output of the power amplifier is $e_a(t)$, the output equation is

$$y = e_a \tag{3.88}$$

Motor and load:

We now find the state-space representation for the motor and load. We could of course use the motor and load block shown in the block diagram on the front endpapers to obtain the result. However, it is more informative to derive the state-space

representation directly from the physics of the motor without first deriving the transfer function. The elements of the derivation were covered in Section 2.8 but are repeated here for continuity. Starting with Kirchhoff's voltage equation around the armature circuit, we find

$$e_a(t) = i_a(t)R_a + K_b\frac{d\theta_m}{dt} \tag{3.89}$$

where $e_a(t)$ is the armature input voltage, $i_a(t)$ is the armature current, R_a is the armature resistance, K_b is the armature constant, and θ_m is the angular displacement of the armature.

The torque, $T_m(t)$, delivered by the motor is related separately to the armature current and the load seen by the armature. From Section 2.8,

$$T_m(t) = K_t i_a(t) = J_m\frac{d^2\theta_m}{dt^2} + D_m\frac{d\theta_m}{dt} \tag{3.90}$$

where J_m is the equivalent inertia as seen by the armature, and D_m is the equivalent viscous damping as seen by the armature.

Solving Eq. (3.90) for $i_a(t)$ and substituting the result into Eq. (3.89) yields

$$e_a(t) = \left(\frac{R_aJ_m}{K_t}\right)\frac{d^2\theta_m}{dt^2} + \left(\frac{D_mR_a}{K_t} + K_b\right)\frac{d\theta_m}{dt} \tag{3.91}$$

Defining the state variables x_1 and x_2 as

$$x_1 = \theta_m \tag{3.92a}$$

$$x_2 = \frac{d\theta_m}{dt} \tag{3.92b}$$

and substituting into Eq. (3.91), we get

$$e_a(t) = \left(\frac{R_aJ_m}{K_t}\right)\frac{dx_2}{dt} + \left(\frac{D_mR_a}{K_t} + K_b\right)x_2 \tag{3.93}$$

Solving for dx_2/dt yields

$$\frac{dx_2}{dt} = -\frac{1}{J_m}\left(D_m + \frac{K_tK_b}{R_a}\right)x_2 + \left(\frac{K_t}{R_aJ_m}\right)e_a(t) \tag{3.94}$$

Using Eqs. (3.92) and (3.94), the state equations are written as

$$\frac{dx_1}{dt} = x_2 \tag{3.95a}$$

$$\frac{dx_2}{dt} = -\frac{1}{J_m}\left(D_m + \frac{K_tK_b}{R_a}\right)x_2 + \left(\frac{K_t}{R_aJ_m}\right)e_a(t) \tag{3.95b}$$

The output, $\theta_o(t)$, is 1/10 the displacement of the armature, which is x_1. Hence, the output equation is

$$y = 0.1x_1 \tag{3.96}$$

In vector-matrix form,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{J_m}\left(D_m + \frac{K_tK_b}{R_a}\right) \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ \frac{K_t}{R_aJ_m} \end{bmatrix}e_a(t) \tag{3.97a}$$

$$y = [0.1 \quad 0]\mathbf{x} \tag{3.97b}$$

But from the case study problem in Chapter 2, $J_m = 0.03$ and $D_m = 0.02$. Also, $K_t/R_a = 0.0625$ and $K_b = 0.5$. Substituting the values into Eq. (3.97a), we obtain the final state-space representation:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -1.71 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 2.083 \end{bmatrix} e_a(t) \tag{3.98a}$$

$$y = [0.1 \quad 0]\mathbf{x} \tag{3.98b}$$

CHALLENGE: You are now given a problem to test your knowledge of this chapter's objectives. Referring to the antenna azimuth position control system shown on the front endpapers, find the state-space representation of each dynamic subsystem. Use Configuration 2.

Pharmaceutical Drug Absorption

An advantage of state-space representation over the transfer function representation is the ability to focus on component parts of a system and write n simultaneous, first-order differential equations rather than attempt to represent the system as a single, nth-order differential equation, as we have done with the transfer function. Also, multiple-input, multiple-output systems can be conveniently represented in state space. This case study demonstrates both of these concepts.

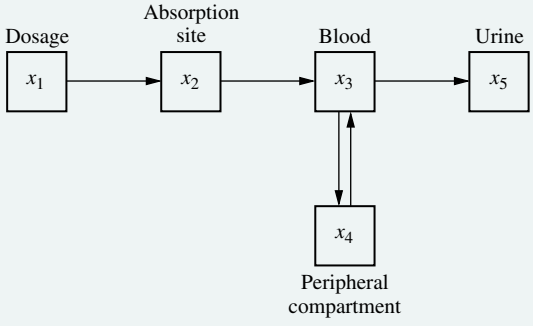


FIGURE 3.16 Pharmaceutical drug-level concentrations in a human

PROBLEM: In the pharmaceutical industry we want to describe the distribution of a drug in the body. A simple model divides the process into compartments: the dosage, the absorption site, the blood, the peripheral compartment, and the urine. The rate of change of the amount of a drug in a compartment is equal to the input flow rate diminished by the output flow rate. Figure 3.16 summarizes the system. Here each x_i is the amount of drug in that particular compartment (*Lordi, 1972*). Represent the system in state space, where the outputs are the amounts of drug in each compartment.

SOLUTION: The flow rate of the drug into any given compartment is proportional to the concentration of the drug in the previous compartment, and the flow rate out of a given compartment is proportional to the concentration of the drug in its own compartment.

We now write the flow rate for each compartment. The dosage is released to the absorption site at a rate proportional to the dosage concentration, or

$$\frac{dx_1}{dt} = -K_1 x_1 \tag{3.99}$$

The flow into the absorption site is proportional to the concentration of the drug at the dosage site. The flow from the absorption site into the blood is proportional to the concentration of the drug at the absorption site. Hence,

$$\frac{dx_2}{dt} = K_1 x_1 - K_2 x_2 \tag{3.100}$$

Similarly, the net flow rate into the blood and peripheral compartment is

$$\frac{dx_3}{dt} = K_2 x_2 - K_3 x_3 + K_4 x_4 - K_5 x_3 \tag{3.101}$$

$$\frac{dx_4}{dt} = K_5 x_3 - K_4 x_4 \tag{3.102}$$

where $(K_4x_4 - K_5x_3)$ is the net flow rate into the blood from the peripheral compartment. Finally, the amount of the drug in the urine is increased as the blood releases the drug to the urine at a rate proportional to the concentration of the drug in the blood. Thus,

$$\frac{dx_5}{dt} = K_3x_3 \tag{3.103}$$

Equations (3.99) through (3.103) are the state equations. The output equation is a vector that contains each of the amounts, x_i. Thus, in vector-matrix form,

$$\dot{\mathbf{x}} = \begin{bmatrix} -K_1 & 0 & 0 & 0 & 0 \\ K_1 & -K_2 & 0 & 0 & 0 \\ 0 & K_2 & -(K_3 + K_5) & K_4 & 0 \\ 0 & 0 & K_5 & -K_4 & 0 \\ 0 & 0 & K_3 & 0 & 0 \end{bmatrix} \mathbf{x} \tag{3.104a}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} \tag{3.104b}$$

You may wonder how there can be a solution to these equations if there is no input. In Chapter 4, when we study how to solve the state equations, we will see that initial conditions will yield solutions without forcing functions. For this problem, an initial condition on the amount of dosage, x_1, will generate drug quantities in all other compartments.

CHALLENGE: We now give you a problem to test your knowledge of this chapter's objectives. The problem concerns the storage of water in aquifers. The principles are similar to those used to model pharmaceutical drug absorption.

Underground water supplies, called aquifers, are used in many areas for agricultural, industrial, and residential purposes. An aquifer system consists of a number of interconnected natural storage tanks. Natural water flows through the sand and sandstone of the aquifer system, changing the water levels in the tanks on its way to the sea. A water conservation policy can be established whereby water is pumped between tanks to prevent its loss to the sea.

A model for the aquifer system is shown in Figure 3.17. In this model, the aquifer is represented by three tanks, with water level h_i called the *head*. Each q_n is the

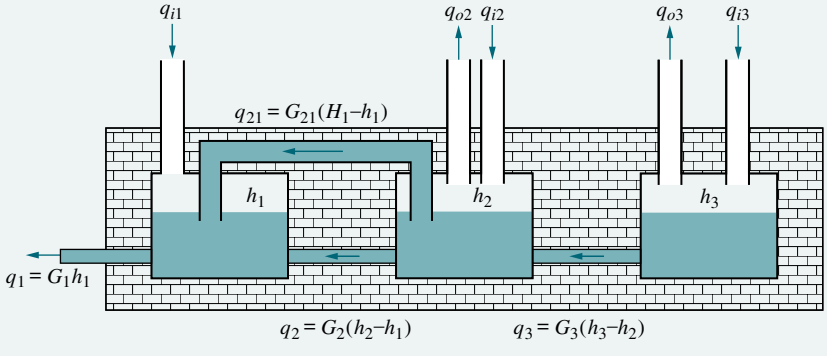


FIGURE 3.17 Aquifer system model

natural water flow to the sea and is proportional to the difference in head between two adjoining tanks, or $q_n = G_n(h_n - h_{n-1})$, where G_n is a constant of proportionality and the units of q_n are m^3/yr.

The engineered flow consists of three components, also measured in m^3/yr: (1) flow from the tanks for irrigation, industry, and homes, q_{on}; (2) replenishing of the tanks from wells, q_{in}; and (3) flow, q_{21}, created by the water conservation policy to prevent loss to the sea. In this model, water for irrigation and industry will be taken only from Tank 2 and Tank 3. Water conservation will take place only between Tank 1 and Tank 2, as follows. Let H_1 be a reference head for Tank 1. If the water level in Tank 1 falls below H_1, water will be pumped from Tank 2 to Tank 1 to replenish the head. If h_1 is higher than H_1, water will be pumped back to Tank 2 to prevent loss to the sea. Calling this *flow for conservation* q_{21}, we can say this flow is proportional to the difference between the head of Tank 1, h_1, and the reference head, H_1, or $q_{21} = G_{21}(H_1 - h_1)$.

The net flow into a tank is proportional to the rate of change of head in each tank. Thus,

$$C_n dh_n/dt = q_{in} - q_{on} + q_{n+1} - q_n + q_{(n+1)n} - q_{n(n-1)}$$

(*Kandel*, *1973*).

Represent the aquifer system in state space, where the state variables and the outputs are the heads of each tank.

Summary

This chapter has dealt with the state-space representation of physical systems, which took the form of a state equation,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.105}$$

and an output equation,

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{3.106}$$

for $t \geq t_0$, and initial conditions $\mathbf{x}(t_0)$. Vector $\mathbf{x}$ is called the *state vector* and contains variables, called *state variables*. The state variables can be combined algebraically with the input to form the output equation, Eq. (3.106), from which any other system variables can be found. State variables, which can represent physical quantities such as current or voltage, are chosen to be linearly independent. The choice of state variables is not unique and affects how the matrices **A, B, C**, and **D** look. We will solve the state and output equations for $\mathbf{x}$ and $\mathbf{y}$ in Chapter 4.

In this chapter, transfer functions were represented in state space. The form selected was the phase-variable form, which consists of state variables that are successive derivatives of each other. In three-dimensional state space, the resulting system matrix, **A**, for the phase-variable representation is of the form

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \tag{3.107}$$

where the a_i's are the coefficients of the characteristic polynomial or denominator of the system transfer function. We also discussed how to convert from a state-space representation to a transfer function.

In conclusion, then, for linear, time-invariant systems, the state-space representation is simply another way of mathematically modeling them. One major advantage of applying the state-space representation to such linear systems is that it allows computer simulation. Programming the system on the digital computer and watching the system's response is an invaluable analysis and design tool. Simulation is covered in Appendix H located at www.wiley.com/college/nise.

Review Questions

1. Give two reasons for modeling systems in state space.
2. State an advantage of the transfer function approach over the state-space approach.
3. Define *state variables.*
4. Define *state.*
5. Define *state vector.*
6. Define *state space.*
7. What is required to represent a system in state space?
8. An eighth-order system would be represented in state space with how many state equations?
9. If the state equations are a system of first-order differential equations whose solution yields the state variables, then the output equation performs what function?
10. What is meant by *linear independence*?
11. What factors influence the choice of state variables in any system?
12. What is a convenient choice of state variables for electrical networks?
13. If an electrical network has three energy-storage elements, is it possible to have a state-space representation with more than three state variables? Explain.
14. What is meant by the phase-variable form of the state-equation?

Problems

1. Represent the electrical network shown in Figure P3.1 in state space, where $v_o(t)$ is the output. [Section: 3.4]

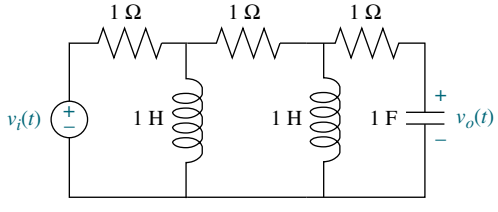


FIGURE P3.1

2. Represent the electrical network shown in Figure P3.2 in state space, where $i_R(t)$ is the output. [Section: 3.4]

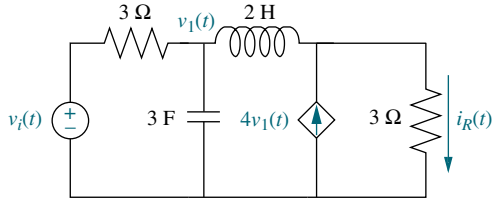


FIGURE P3.2

3. Find the state-space representation of the network shown in Figure P3.3 if the output is $v_o(t)$. [Section: 3.4]

WileyPLUS WPCS Control Solutions

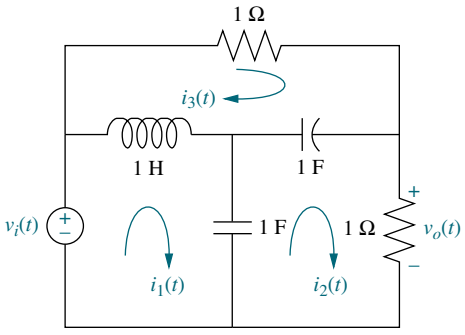


FIGURE P3.3

4. Represent the system shown in Figure P3.4 in state space where the output is $x_3(t)$. [Section: 3.4]

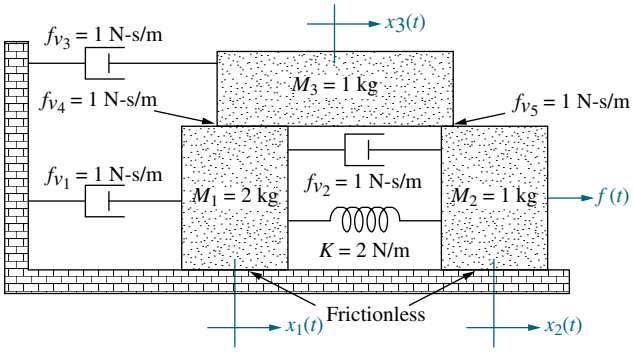


FIGURE P3.4

5. Represent the translational mechanical system shown in Figure P3.5 in state space, where $x_1(t)$ is the output. [Section: 3.4]

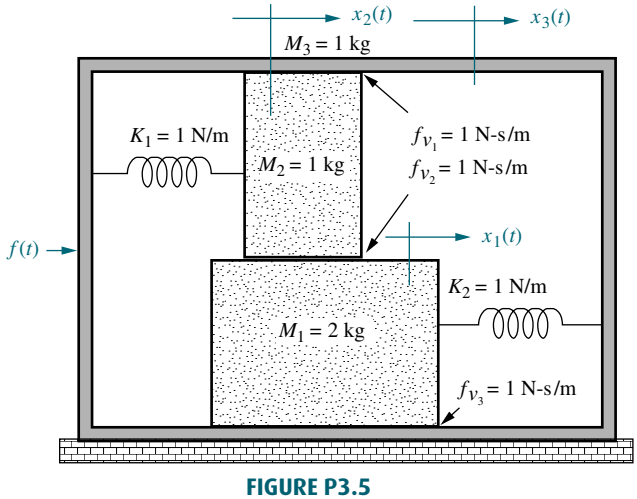


FIGURE P3.5

6. Represent the rotational mechanical system shown in Figure P3.6 in state space, where $\theta_1(t)$ is the output. [Section: 3.4]

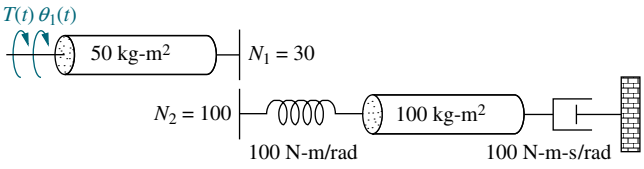


FIGURE P3.6

7. Represent the system shown in Figure P3.7 in state space where the output is $\theta_L(t)$. [Section: 3.4]

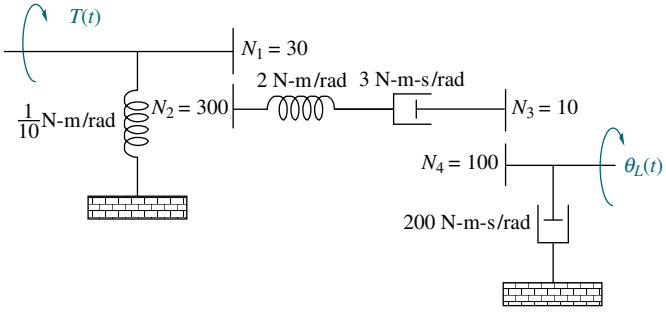


FIGURE P3.7

8. Show that the system of Figure 3.7 in the text yields a fourth-order transfer function if we relate the displacement of either mass to the applied force, and a third-order one if we relate the velocity of either mass to the applied force. [Section: 3.4]

9. Find the state-space representation in phase-variable form for each of the systems shown in Figure P3.8. [Section: 3.5]

WileyPLUS WPCS Control Solutions

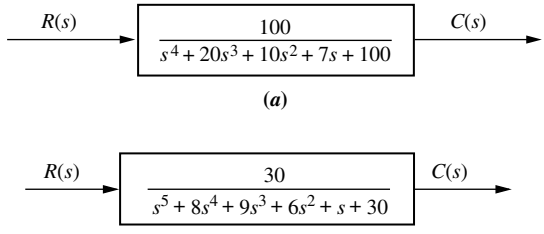


FIGURE P3.8

10. Repeat Problem 9 using MATLAB. [Section: 3.5]

MATLAB ML

11. For each system shown in Figure P3.9, write the state equations and the output equation for the phase-variable representation. [Section: 3.5]

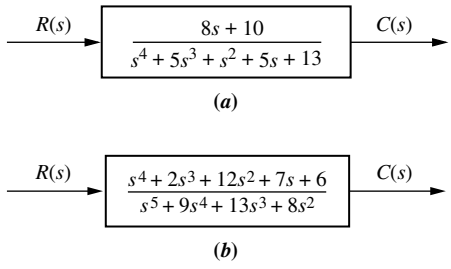


FIGURE P3.9

12. Repeat Problem 11 using MATLAB. [Section: 3.5]

MATLAB ML

13. Represent the following transfer function in state space. Give your answer in vector-matrix form. [Section: 3.5]

$$T(s) = \frac{(s^2 + 3s + 8)}{(s+1)(s^2 + 5s + 5)}$$

14. Find the transfer function $G(s) = Y(s)/R(s)$ for each of the following systems represented in state space: [Section: 3.6]

WileyPLUS WPCS Control Solutions

a.
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -2 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} r$$

$$y = [1 \quad 0 \quad 0]\mathbf{x}$$

b.
$$\dot{\mathbf{x}} = \begin{bmatrix} 2 & -3 & -8 \\ 0 & 5 & 3 \\ -3 & -5 & -4 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix} r$$

$$y = [1 \quad 3 \quad 6]\mathbf{x}$$

c.
$$\dot{\mathbf{x}} = \begin{bmatrix} 3 & -5 & 2 \\ 1 & -8 & 7 \\ -3 & -6 & 2 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 5 \\ -3 \\ 2 \end{bmatrix} r$$

$$y = [1 \quad -4 \quad 3]\mathbf{x}$$

15. Use MATLAB to find the transfer function, $G(s) = Y(s)/R(s)$, for each of the following systems represented in state space: [Section: 3.6]

MATLAB ML

a.
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -7 & -9 & -2 & -3 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 5 \\ 8 \\ 2 \end{bmatrix} r$$

$$y = [1 \quad 3 \quad 6 \quad 6]\mathbf{x}$$

b.
$$\dot{\mathbf{x}} = \begin{bmatrix} 3 & 1 & 0 & 4 & -2 \\ -3 & 5 & -5 & 2 & -1 \\ 0 & 1 & -1 & 2 & 8 \\ -7 & 6 & -3 & -4 & 0 \\ -6 & 0 & 4 & -3 & 1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 2 \\ 7 \\ 8 \\ 5 \\ 4 \end{bmatrix} r$$

$$y = [1 \quad -2 \quad -9 \quad 7 \quad 6]\mathbf{x}$$

16. Repeat Problem 15 using MATLAB, the Symbolic Math Toolbox, and Eq. (3.73). [Section: 3.6]

Symbolic Math SM

17. Gyros are used on space vehicles, aircraft, and ships for inertial navigation. The gyro shown in Figure P3.10 is a rate gyro restrained by springs connected between the inner gimbal and the outer gimbal (frame) as shown. A rotational rate about the z-axis causes the rotating disk to precess about the x-axis. Hence, the input is a rotational rate about the z-axis, and the output is an angular displacement about the x-axis. Since the outer gimbal is secured to the vehicle, the displacement about the x-axis is a measure of the vehicle's angular rate about the z-axis. The equation of motion is

WileyPLUS WPCS Control Solutions

$$J_x \frac{d^2\theta_x}{dt^2} + D_x \frac{d\theta_x}{dt} + K_x\theta_x = J\omega \frac{d\theta_z}{dt}$$

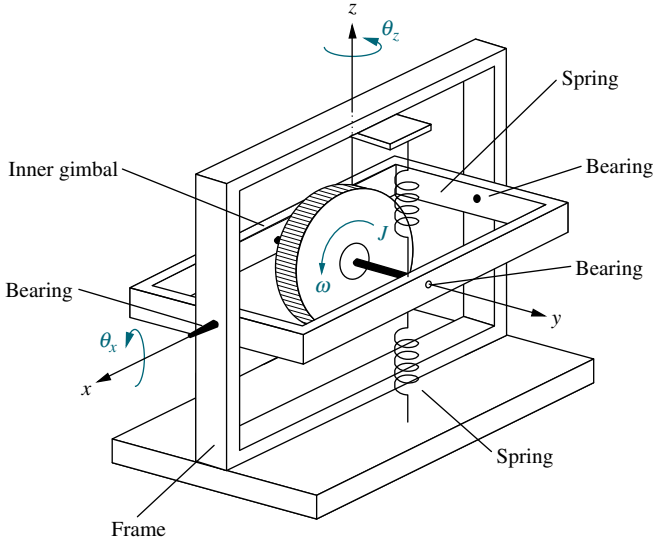


FIGURE P3.10 Gyro system

Represent the gyro in state space. [Section: 3.4]

18. A missile in flight, as shown in Figure P3.11, is subject to several forces: thrust, lift, drag, and gravity. The missile flies at an angle of attack, α, from its longitudinal axis, creating lift. For steering, the body angle from vertical, ϕ, is controlled by rotating the engine at the tail. The transfer function relating the body angle, ϕ, to the angular displacement, δ, of the engine is of the form

$$\frac{\Phi(s)}{\delta(s)} = \frac{K_a s + K_b}{K_3 s^3 + K_2 s^2 + K_1 s + K_0}$$

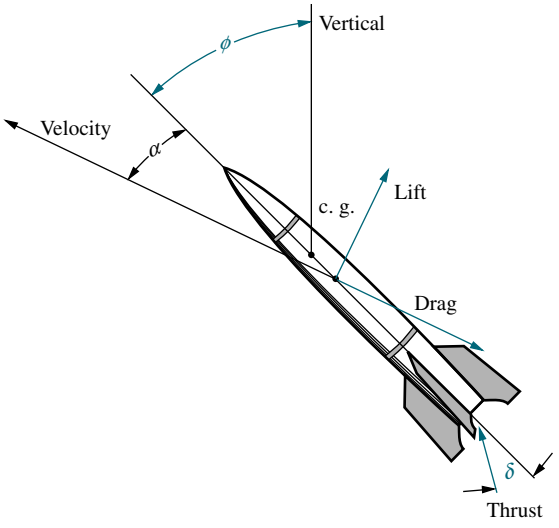


FIGURE P3.11 Missile

Represent the missile steering control in state space. [Section: 3.5]

19. Given the dc servomotor and load shown in Figure P3.12, represent the system in state space, where the state variables are the armature current, i_a, load displacement, θ_L, and load angular velocity, ω_L. Assume that the output is the angular displacement of the armature. Do not neglect armature inductance. [Section: 3.4]

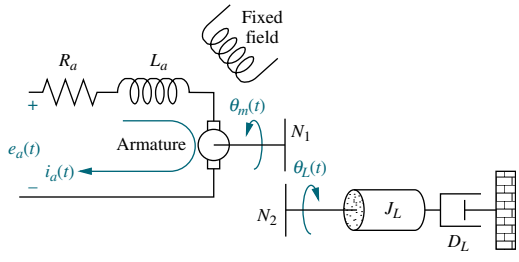


FIGURE P3.12 Motor and load

20. Consider the mechanical system of Figure P3.13. If the spring is nonlinear, and the force, F_s, required to stretch the spring is $F_s = 2x_1^2$, represent the system in state space linearized about $x_1 = 1$ if the output is x_2. [Section: 3.7]

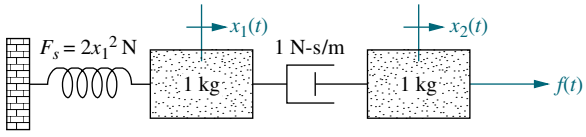


FIGURE P3.13 Nonlinear mechanical system

21. Image-based homing for robots can be implemented by generating heading command inputs to a steering system based on the following guidance algorithm. Suppose the robot shown in Figure P3.14(*a*) is to go from point *R* to a target, point *T*, as shown in Figure P3.14(*b*). If $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$ are vectors from the robot to each landmark, *X*, *Y*, *Z*, respectively, and $\mathbf{T}_x$, $\mathbf{T}_y$, and $\mathbf{T}_z$ are vectors from the target to each landmark, respectively, then heading commands would drive the robot to minimize $\mathbf{R}_x - \mathbf{T}_x$, $\mathbf{R}_y - \mathbf{T}_y$, and $\mathbf{R}_z - \mathbf{T}_z$ simultaneously, since the differences will be zero when the robot arrives at the target (*Hong, 1992*). If Figure P3.14(*c*) represents the control system that steers the robot, represent each block—the controller, wheels, and vehicle—in state space. [Section: 3.5]

WileyPLUS WPCS Control Solutions

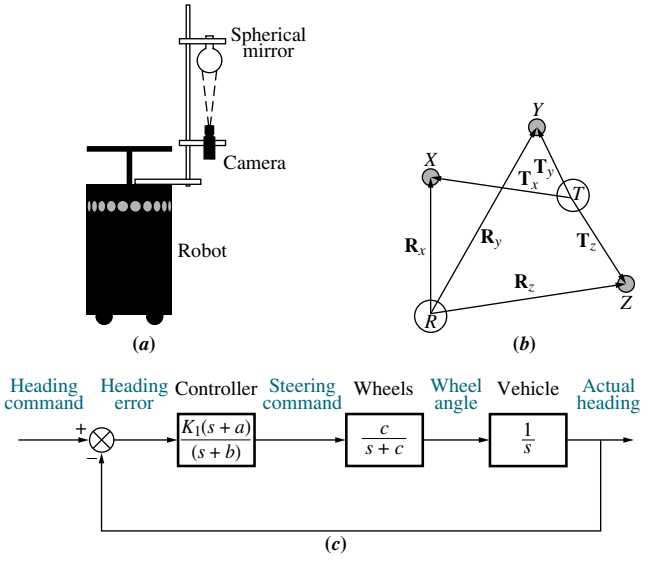


FIGURE P3.14 **a.** Robot with television imaging system; **b.** vector diagram showing concept behind image-based homing; **c.** heading control system (© 1992 IEEE)

22. Given the F4-E military aircraft shown in Figure P3.15(*a*), where normal acceleration, a_n, and pitch

rate, q, are controlled by elevator deflection, δ_e, on the horizontal stabilizers and by canard deflection, δ_e. A commanded deflection, δ_{com}, as shown in Figure P3.15(b), is used to effect a change in both δ_e and δ_c. The relationships are

$$\frac{\delta_e(s)}{\delta_{\text{com}}(s)} = \frac{1/\tau}{s+1/\tau}$$
$$\frac{\delta_c(s)}{\delta_{\text{com}}(s)} = \frac{K_c/\tau}{s+1/\tau}$$

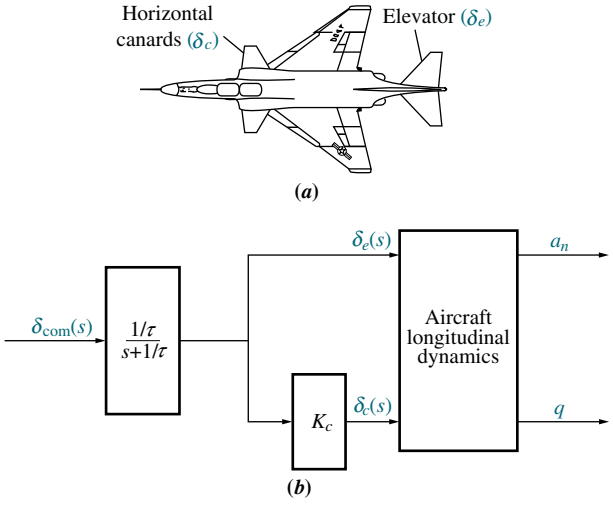


FIGURE P3.15 **a.** F4-E with canards **b.** open-loop flight control system

These deflections yield, via the aircraft longitudinal dynamics, a_n and q. The state equations describing the effect of δ_{com} on a_n and q is given by (*Cavallo*, *1992*)

$$\begin{bmatrix} \dot{a}_n \\ \dot{q} \\ \dot{\delta}_e \end{bmatrix} = \begin{bmatrix} -1.702 & 50.72 & 263.38 \\ 0.22 & -1.418 & -31.99 \\ 0 & 0 & -14 \end{bmatrix} \begin{bmatrix} a_n \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} -272.06 \\ 0 \\ 14 \end{bmatrix} \delta_{\text{com}}$$

Find the following transfer functions: [Section: 3.5]

$$G_1(s) = \frac{A_n(s)}{\delta_{\text{com}}(s)}$$
$$G_2(s) = \frac{Q(s)}{\delta_{\text{com}}(s)}$$

23. Modern robotic manipulators that act directly upon their target environments must be controlled so that impact forces as well as steady-state forces do not damage the targets. At the same time, the manipulator must provide sufficient force to perform the task. In order to develop a control system to regulate these forces, the robotic manipulator and target environment must be modeled. Assuming the model shown in Figure P3.16, represent in state space the manipulator and its environment under the following conditions (*Chiu*, *1997*). [Section: 3.5]

a. The manipulator is not in contact with its target environment.

b. The manipulator is in constant contact with its target environment.

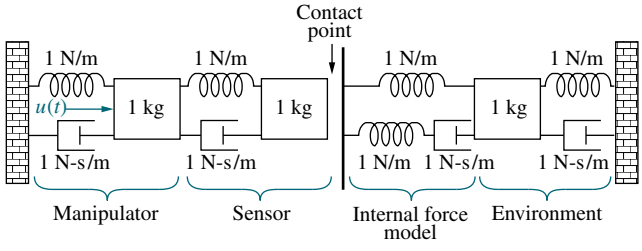


FIGURE P3.16 Robotic manipulator and target environment (© 1997 IEEE)

24. In the past, Type-1 diabetes patients had to inject themselves with insulin three to four times a day. New delayed-action insulin analogues such as insulin Glargine require a single daily dose. A similar procedure to the one described in the Pharmaceutical Drug Absorption case study of this chapter is used to find a model for the concentration-time evolution of plasma for insulin Glargine. For a specific patient, state-space model matrices are given by (*Tarín*, *2007*)

$$\mathbf{A} = \begin{bmatrix} -0.435 & 0.209 & 0.02 \\ 0.268 & -0.394 & 0 \\ 0.227 & 0 & -0.02 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};$$

$$\mathbf{C} = [0.0003\ 0\ 0]; \quad \mathbf{D} = 0$$

where the state vector is given by

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

The state variables are

x_1 = insulin amount in plasma compartment
x_2 = insulin amount in liver compartment
x_3 = insulin amount in interstitial (in body tissue) compartment

The system's input is u = external insulin flow. The system's output is y = plasma insulin concentration.

a. Find the system's transfer function.

MATLAB

b. `Verify your result using MATLAB.` ML

25. A linear, time-invariant model of the hypothalamic-pituitary-adrenal axis of the endocrine system with five state variables has been proposed as follows (*Kyrylov, 2005*):

$$\frac{dx_0}{dt} = a_{00}x_0 + a_{02}x_2 + d_0$$
$$\frac{dx_1}{dt} = a_{10}x_0 + a_{11}x_1 + a_{12}x_2$$
$$\frac{dx_2}{dt} = a_{20}x_0 + a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4$$
$$\frac{dx_3}{dt} = a_{32}x_2 + a_{33}x_3$$
$$\frac{dx_4}{dt} = a_{42}x_2 + a_{44}x_4$$

where each of the state variables represents circulatory concentrations as follows:

x_0 = corticotropin-releasing hormone
x_1 = corticotropin
x_2 = free cortisol
x_3 = albumin-bound cortisol
x_4 = corticosteroid-binding globulin
d_0 = an external generating factor

Express the system in the form $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$.

26. In this chapter, we described the state-space representation of single-input, single-output systems. In general, systems can have multiple inputs and multiple outputs. An autopilot is to be designed for a submarine as shown in Figure P3.17 to maintain a constant depth under severe wave disturbances. We will see that this system has two inputs and two outputs and thus the scaler u becomes a vector, $\mathbf{u}$, and the scaler y becomes a vector, $\mathbf{y}$, in the state equations.

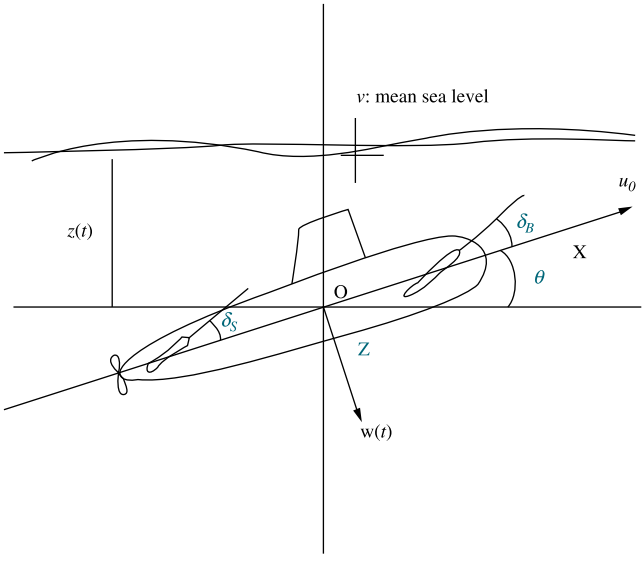


FIGURE P3.17 (© 1995 IEEE)

It has been shown that the system's linearized dynamics under neutral buoyancy and at a given constant speed are given by (*Liceaga-Castro, 2009*):

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\mathbf{y} = \mathbf{Cx}$$

where

$$\mathbf{x} = \begin{bmatrix} w \\ q \\ z \\ \theta \end{bmatrix}; \quad \mathbf{y} = \begin{bmatrix} z \\ \theta \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \delta_B \\ \delta_S \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -0.038 & 0.896 & 0 & 0.0015 \\ 0.0017 & -0.092 & 0 & -0.0056 \\ 1 & 0 & 0 & -3.086 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

$$\mathbf{B} = \begin{bmatrix} -0.0075 & -0.023 \\ 0.0017 & -0.0022 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and where

w = the heave velocity
q = the pitch rate
z = the submarine depth
θ = the pitch angle
δ_B = the bow hydroplane angle
δ_S = the stern hydroplane angle

Since this system has two inputs and two outputs, four transfer functions are possible.

a. Use MATLAB to calculate the system's matrix transfer function. MATLAB ML

b. Using the results from Part **a**, write the transfer function $\frac{z(s)}{\delta_B(s)}$, $\frac{z(s)}{\delta_S(s)}$, $\frac{\theta(s)}{\delta_B(s)}$, and $\frac{\theta(s)}{\delta_S(s)}$.

27. Experiments to identify precision grip dynamics between the index finger and thumb have been performed using a ball-drop experiment. A subject holds a device with a small receptacle into which an object is dropped, and the response is measured (*Fagergren, 2000*). Assuming a step input, it has been found that the response of the motor subsystem together with the sensory system is of the form

$$G(s) = \frac{Y(s)}{R(s)} = \frac{s + c}{(s^2 + as + b)(s + d)}$$

Convert this transfer function to a state-space representation.

28. State-space representations are, in general, not unique. One system can be represented in several possible ways. For example, consider the following systems:

a. $\dot{x} = -5x + 3u$

$y = 7x$

b.
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -5 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 7 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

c.
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -5 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 7 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Show that these systems will result in the same transfer function. We will explore this phenomenon in more detail in Chapter 5.

29. Figure P3.18 shows a schematic description of the global carbon cycle (*Li, 2009*). In the figure, $m_A(t)$ represents the amount of carbon in gigatons (GtC) present in the atmosphere of earth; $m_V(t)$ the amount in vegetation; $m_s(t)$ the amount in soil; $m_{SO}(t)$ the amount in surface ocean; and $m_{IDO}(t)$ the amount in intermediate and deep-ocean reservoirs. Let $u_E(t)$ stand for the human generated CO_2 emissions (GtC/yr). From the figure, the atmospheric mass balance in the atmosphere can be expressed as:

$$\frac{dm_A}{dt}(t) = u_E(t) - (k_{O1} + k_{L1})m_A(t) + k_{L2}m_V(t) + k_{O2}m_{SO}(t) + k_{L4}m_S(t)$$

where the k's are exchange coefficients (yr^{-1}).

a. Write the remaining reservoir mass balances. Namely, write equations for $\frac{dm_{SO}(t)}{dt}$, $\frac{dm_{IDO}(t)}{dt}$, $\frac{dm_V(t)}{dt}$, and $\frac{dm_S(t)}{dt}$

b. Express the system in state space form.

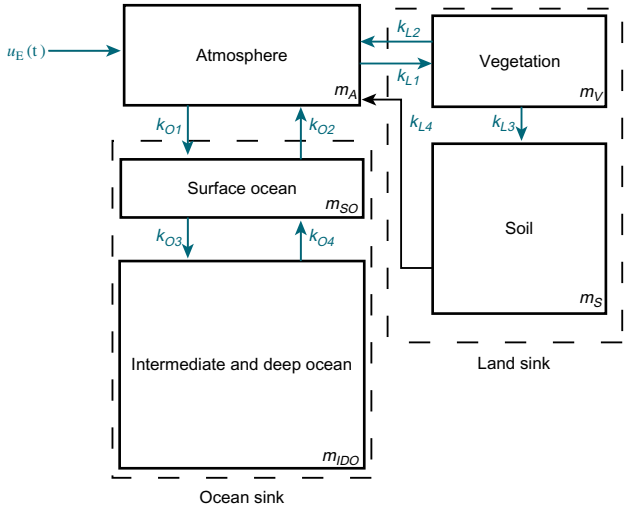


FIGURE P3.18 Global carbon cycle

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

30. High-speed rail pantograph. A translational mechanical system model for a high-speed rail pantograph, used to supply electricity to a train from an overhead catenary, is shown in Figure P2.39(*b*) (*O'Connor, 1979*). Represent the pantograph in

state space, where the output is the displacement of the top of the pantograph, $y_h(t) - y_{\text{cat}}(t)$.

31. Control of HIV/AIDS. Problem 68 in Chapter 2 introduced a model for HIV infection. If retroviral drugs, RTIs and PIs as discussed in Problem 22 in Chapter 1, are used, the model is modified as follows (*Craig, 2004*):

$$\frac{dT}{dt} = s - dT - (1 - u_1)\beta T v$$
$$\frac{dT^*}{dt} = (1 - u_1)\beta T v - \mu T^*$$
$$\frac{dv}{dt} = (1 - u_2)kT^* - cv$$

where $0 \le u_1 \le 1, 0 \le u_2 \le 1$ represent the effectiveness of the RTI and PI medication, respectively.

a. Obtain a state-space representation of the HIV/AIDS model by linearizing the equations about the

$$(T_0,\ T_0^*,\ v_0) = \left(\frac{c\mu}{\beta k}, \frac{s}{\mu} - \frac{cd}{\beta k}, \frac{sk}{c\mu} - \frac{d}{\beta}\right)$$

equilibrium with $u_{10} = u_{20} = 0$. This equilibrium represents the asymptomatic HIV-infected patient. Note that each one of the above equations is of the form $\dot{x}_i = f_i(x_i, u_1, u_2) i = 1, 2, 3$.

b. If Matrices **A** and **B** are given by

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{T_0, T_0^*, v_0}; \qquad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix}_{T_0, T_0^*, v_0}$$

and we are interested in the number of free HIV viruses as the system's output,

$$\mathbf{C} = [0 \quad 0 \quad 1]$$

show that

$$\mathbf{A} = \begin{bmatrix} -(d + \beta v_0) & 0 & -\beta T_0 \\ \beta v_0 & -\mu & \beta T_0 \\ 0 & k & -c \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} \beta T_0 v_0 & 0 \\ -\beta T_0 v_0 & 0 \\ 0 & -kT_0^* \end{bmatrix}$$

c. Typical parameter values and descriptions for the HIV/AIDS model are shown in the following table.

t	Time	days
d	Death of uninfected T cells	0.02/day
k	Rate of free viruses produced per infected T cell	100 counts/cell
s	Source term for uninfected T cells	$10/\text{mm}^3/\text{day}$
β	Infectivity rate of free virus particles	$2.4 \times 10^{-5}/\text{mm}^3/\text{day}$
c	Death rate of viruses	2.4/day
μ	Death rate of infected T cells	0.24/day

(© 2004 IEEE)

Substitute the values from the table into your model and write as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

32. Hybrid vehicle. For Problem 23 in Chapter 1 we developed the functional block diagrams for the cruise control of serial, parallel, and split-power hybrid electric vehicles (HEV). Those diagrams showed that the engine or electric motor or both may propel the vehicle. When electric motors are the sole providers of the motive force, the forward paths of all HEV topologies are similar. In general, such a forward path can be represented (*Preitl, 2007*) by a block diagram similar to the one of Figure P3.19.

Assume the motor to be an armature-controlled dc motor. In this diagram, K_A is the power amplifier gain; $G_e(s)$ is the transfer function of the motor electric circuit and consists of a series inductor and resistor, L_a and R_a, respectively; K_t is the motor torque constant; J_{tot}, is the sum of the motor inertia, J_m, the inertias of the vehicle, J_{veh}, and the two driven wheels, J_w, both of which are reflected to the motor shaft; k_f is the coefficient of viscous friction; and k_b is the back emf constant.

The input variables are $u_c(t)$, the command voltage from the electronic control unit and $T_c(t)$, the load torque. The output variables in this block diagram are the motor angular speed, $\omega(t)$, and its armature current, $I_a(t)$.

a. Write the basic time-domain equations that characterize the relationships between the state, input, and output variables for the block diagram of Figure P3.19, given that the state variables are the motor armature current, $I_a(t)$, and angular speed, $\omega(t)$.

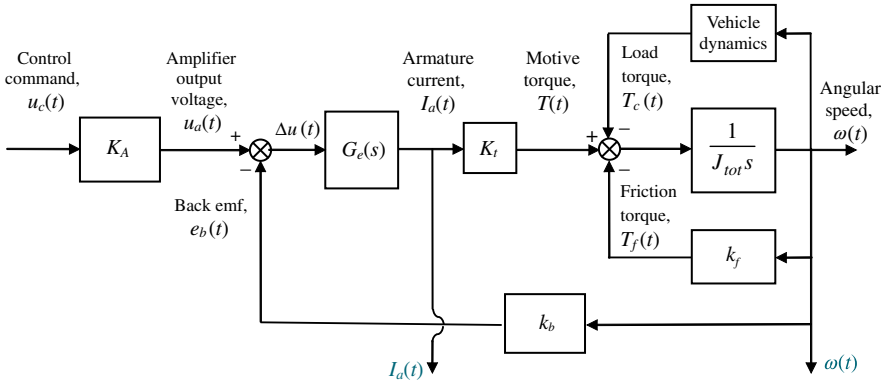


FIGURE P3.19 Block diagram representation of an HEV forward path (© 2007 IEEE)

b. Write the resulting state-space equations and then represent them in matrix form. Regard the load torque $T_c(t)$ as an extra input to the system. Thus, in your resulting state-space representation, the system will have two inputs and two outputs.

Cyber Exploration Laboratory

Experiment 3.1

Objectives To learn to use MATLAB to (1) generate an LTI state-space representation of a system and (2) convert an LTI state-space representation of a system to an LTI transfer function.

Minimum Required Software Packages MATLAB and the Control System Toolbox

Prelab

1. Derive the state-space representation of the translational mechanical system shown in Skill-Assessment Exercise 3.2 if you have not already done so. Consider the output to be $x_3(t)$.
2. Derive the transfer function, $\frac{X_3(s)}{F(s)}$, from the equations of motion for the translational mechanical system shown in Skill-Assessment Exercise 3.2.

Lab

1. Use MATLAB to generate the LTI state-space representation derived in Prelab 1.
2. Use MATLAB to convert the LTI state-space representation found in Lab 1 to the LTI transfer function found in Prelab 2.

Postlab

1. Compare your transfer functions as found from Prelab 2 and Lab 2.
2. Discuss the use of MATLAB to create LTI state-space representations and the use of MATLAB to convert these representations to transfer functions.

Experiment 3.2

Objectives To learn to use MATLAB and the Symbolic Math Toolbox to (1) find a symbolic transfer function from the state-space representation and (2) find a state-space representation from the equations of motion.

Minimum Required Software Packages MATLAB, the Symbolic Math Toolbox, and the Control System Toolbox

Prelab

1. Perform Prelab 1 and Prelab 2 of Experiment 3.1 if you have not already done so.
2. Using the equation $T(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ to find a transfer function from a state-space representation, write a MATLAB program using the Symbolic Math Toolbox to find the symbolic transfer function from the state-space representation of the translational mechanical system shown in Skill-Assessment Exercise 3.2 and found as a step in Prelab 1.
3. Using the equations of motion of the translational mechanical system shown in Skill-Assessment Exercise 3.2 and found in Prelab 1, write a symbolic MATLAB program to find the transfer function, $\frac{X_3(s)}{F(s)}$, for this system.

Lab

1. Run the programs composed in Prelabs 2 and Prelab 3 and obtain the symbolic transfer functions by the two methods.

Postlab

1. Compare the symbolic transfer function obtained from $T(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ with the symbolic transfer function obtained from the equations of motion.
2. Discuss the advantages and disadvantages between the two methods.
3. Describe how you would obtain an LTI state-space representation and an LTI transfer function from your symbolic transfer function.

Experiment 3.3

Objectives To learn to use LabVIEW to (1) generate state-space representations of transfer functions, (2) generate transfer functions from state-space representations, and (3) verify that there are multiple state-space representations for a transfer function.

Minimum Required Software Packages LabVIEW, the LabVIEW Control Design and Simulation Module, and the MathScript RT Module.

Prelab

1. Study Appendix D, Sections D.1 through Section D.4, Example D.1.
2. Solve Skill-Assessment Exercise 3.3 in Chapter 3.
3. Use your solution to Prelab 2 and convert back to the transfer function.

Lab

1. Use LabVIEW to convert the transfer function, $G(s) = \dfrac{2s+1}{s^2+7s+9}$, into a state-space representation using both the graphical and MathScript approaches. The front panel will contain controls for the entry of the transfer function and indicators of the transfer function and the two state-space results. Functions for this experiment can be found in the following palettes: (1) **Control Design and Simulation/Control Design/Model Construction,** (2) **Control Design and Simulation/Control Design/Model Conversion,** and (3) **Programming/Structures** Hint: Coefficients are entered in reverse order when using MathScript with MATLAB.
2. Use LabVIEW to convert all state-space representations found in Lab 1 to a transfer function. All state-space conversions should yield the transfer function given in Lab 1. The front panel will contain controls for entering state-space representations and indicators of the transfer function results as well as the state equations used.

Postlab

1. Describe any correlation found between the results of Lab 1 and calculations made in the Prelab.
2. Describe and account for any differences between the results of Lab 1 and calculations made in the Prelab.
3. Explain the results of Lab 2 and draw conclusions from the results.

Bibliography

Carlson, L. E., and Griggs, G. E. *Aluminum Catenary System Quarterly Report.* Technical Report Contract Number DOT-FR-9154, U.S. Department of Transportation, 1980.

Cavallo, A., De Maria, G., and Verde, L. Robust Flight Control Systems: A Parameter Space Design. *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, September–October 1992, pp. 1207–1215.

Cereijo, M. R. State Variable Formulations. *Instruments and Control Systems*, December 1969, pp. 87–88.

Chiu, D. K., and Lee, S. Design and Experimentation of a Jump Impact Controller. *IEEE Control Systems*, June 1997, pp. 99–106.

Cochin, I. *Analysis and Design of Dynamic Systems*. Harper & Row, New York, 1980.

Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

Elkins, J. A. *A Method for Predicting the Dynamic Response of a Pantograph Running at Constant Speed under a Finite Length of Overhead Equipment*. Technical Report TN DA36, British Railways, 1976.

Fagergren, A., Ekeberg, Ö., and Forssberg, H. Precision Grip Force Dynamics: A System Identification Approach. *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 10, 2000, pp. 1366–1375.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems.* Addison-Wesley, Reading, MA, 1986.

Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E. M. Image-Based Homing. *IEEE Control Systems*, February 1992, pp. 38–45.

Inigo, R. M. Observer and Controller Design for D.C. Positional Control Systems Using State Variables. *Transactions, Analog/Hybrid Computer Educational Society*, December 1974, pp. 177–189.

Kailath, T. *Linear Systems*. Prentice Hall, Upper Saddle River, NJ, 1980.

Kandel, A. Analog Simulation of Groundwater Mining in Coastal Aquifers. *Transactions, Analog/Hybrid Computer Educational Society*, November 1973, pp. 175–183.

Kyrylov, V., Severyanova, L. A., and Vieira, A. Modeling Robust Oscillatory Behavior of the Hypothalamic-Pituitary-Adrenal Axis. *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 12, 2005, pp. 1977–1983.

Li, S., Jarvis, A.J., and Leedal, D.T., Are Response Function Representations of the Global Carbon Cycle Ever Interpretable? *Tellus*, vol. 61B, 2009, pp. 361–371.

Liceaga-Castro, E., van der Molen, G. M. Submarine H^{∞} Depth Control Under Wave Disturbances. *IEEE Transactions on Control Systems Technology*, vol. 3, no. 3, 1995, pp. 338–346.

Lordi, N. G. Analog Computer Generated Lecture Demonstrations in Pharmacokinetics. *Transactions, Analog/Hybrid Computer Educational Society*, November 1972, pp. 217–222.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. 1–4.

Philco Technological Center. *Servomechanism Fundamentals and Experiments*. Prentice Hall, Upper Saddle River, NJ, 1980.

Preitl, Z., Bauer, P., and Bokor, J. A Simple Control Solution for Traction Motor Used in Hybrid Vehicles. *Fourth International Symposium on Applied Computational Intelligence and Informatics. IEEE*. 2007.

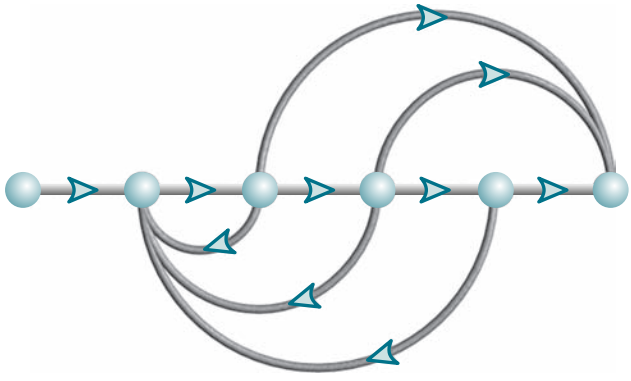
Riegelman, S. et al. Shortcomings in Pharmacokinetic Analysis by Conceiving the Body to Exhibit Properties of a Single Compartment. *Journal of Pharmaceutical Sciences*, vol. 57, no. 1, 1968, pp. 117–123.

Tarin, C., Teufel, E., Picó, J., Bondia, J., and Pfleiderer, H. J. Comprehensive Pharmacokinetic Model of Insulin Glargine and Other Insulin Formulations. *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 12, 2005, pp. 1994–2005.

Timothy, L. K., and Bona, B. E. *State Space Analysis: An Introduction*. McGraw-Hill, New York, 1968.

Time Response

4



Chapter Learning Outcomes

After completing this chapter the student will be able to:

- Use poles and zeros of transfer functions to determine the time response of a control system (Sections 4.1–4.2)
- Describe quantitatively the transient response of first-order systems (Section 4.3)
- Write the general response of second-order systems given the pole location (Section 4.4)
- Find the damping ratio and natural frequency of a second-order system (Section 4.5)
- Find the settling time, peak time, percent overshoot, and rise time for an underdamped second-order system (Section 4.6)
- Approximate higher-order systems and systems with zeros as first- or second-order systems (Sections 4.7–4.8)
- Describe the effects of nonlinearities on the system time response (Section 4.9)
- Find the time response from the state-space representation (Sections 4.10–4.11)

Case Study Learning Outcomes

You will be able to demonstrate your knowledge of the chapter objectives with case studies as follows:

- Given the antenna azimuth position control system shown on the front endpapers, you will be able to (1) predict, by inspection, the form of the open-loop angular velocity response of the load to a step voltage input to the power amplifier;

(2) describe quantitatively the transient response of the open-loop system; (3) derive the expression for the open-loop angular velocity output for a step voltage input; (4) obtain the open-loop state-space representation; (5) plot the open-loop velocity step response using a computer simulation.

- Given the block diagram for the Unmanned Free-Swimming Submersible (UFSS) vehicle's pitch control system shown on the back endpapers, you will be able to predict, find, and plot the response of the vehicle dynamics to a step input command. Further, you will be able to evaluate the effect of system zeros and higher-order poles on the response. You also will be able to evaluate the roll response of a ship at sea.

4.1 Introduction

In Chapter 2, we saw how transfer functions can represent linear, time-invariant systems. In Chapter 3, systems were represented directly in the time domain via the state and output equations. After the engineer obtains a mathematical representation of a subsystem, the subsystem is analyzed for its transient and steady-state responses to see if these characteristics yield the desired behavior. This chapter is devoted to the analysis of system transient response.

It may appear more logical to continue with Chapter 5, which covers the modeling of closed-loop systems, rather than to break the modeling sequence with the analysis presented here in Chapter 4. However, the student should not continue too far into system representation without knowing the application for the effort expended. Thus, this chapter demonstrates applications of the system representation by evaluating the transient response from the system model. Logically, this approach is not far from reality, since the engineer may indeed want to evaluate the response of a subsystem prior to inserting it into the closed-loop system.

After describing a valuable analysis and design tool, poles and zeros, we begin analyzing our models to find the step response of first- and second-order systems. The order refers to the order of the equivalent differential equation representing the system—the order of the denominator of the transfer function after cancellation of common factors in the numerator or the number of simultaneous first-order equations required for the state-space representation.

4.2 Poles, Zeros, and System Response

The output response of a system is the sum of two responses: the *forced response* and the *natural response*.[1] Although many techniques, such as solving a differential equation or taking the inverse Laplace transform, enable us to evaluate this output response, these techniques are laborious and time-consuming. Productivity is aided by analysis and design techniques that yield results in a minimum of time. If the technique is so rapid that we feel we derive the desired result by inspection, we sometimes use the attribute *qualitative* to describe the method. The use of poles and

[1] The forced response is also called the *steady-state response* or *particular solution*. The natural response is also called the *homogeneous solution*.

zeros and their relationship to the time response of a system is such a technique. Learning this relationship gives us a qualitative "handle" on problems. The concept of poles and zeros, fundamental to the analysis and design of control systems, simplifies the evaluation of a system's response. The reader is encouraged to master the concepts of poles and zeros and their application to problems throughout this book. Let us begin with two definitions.

Poles of a Transfer Function

The *poles* of a transfer function are (1) the values of the Laplace transform variable, s, that cause the transfer function to become infinite or (2) any roots of the denominator of the transfer function that are common to roots of the numerator.

Strictly speaking, the poles of a transfer function satisfy part (1) of the definition. For example, the roots of the characteristic polynomial in the denominator are values of s that make the transfer function infinite, so they are thus poles. However, if a factor of the denominator can be canceled by the same factor in the numerator, the root of this factor no longer causes the transfer function to become infinite. In control systems, we often refer to the root of the canceled factor in the denominator as a pole even though the transfer function will not be infinite at this value. Hence, we include part (2) of the definition.

Zeros of a Transfer Function

The *zeros* of a transfer function are (1) the values of the Laplace transform variable, s, that cause the transfer function to become zero, or (2) any roots of the numerator of the transfer function that are common to roots of the denominator.

Strictly speaking, the zeros of a transfer function satisfy part (1) of this definition. For example, the roots of the numerator are values of s that make the transfer function zero and are thus zeros. However, if a factor of the numerator can be canceled by the same factor in the denominator, the root of this factor no longer causes the transfer function to become zero. In control systems, we often refer to the root of the canceled factor in the numerator as a zero even though the transfer function will not be zero at this value. Hence, we include part (2) of the definition.

Poles and Zeros of a First-Order System: An Example

Given the transfer function $G(s)$ in Figure 4.1(a), a pole exists at $s = -5$, and a zero exists at -2. These values are plotted on the complex s-plane in Figure 4.1(b), using an $\times$ for the pole and a $\bigcirc$ for the zero. To show the properties of the poles and zeros, let us find the unit step response of the system. Multiplying the transfer function of Figure 4.1(a) by a step function yields

$$C(s) = \frac{(s+2)}{s(s+5)} = \frac{A}{s} + \frac{B}{s+5} = \frac{2/5}{s} + \frac{3/5}{s+5} \tag{4.1}$$

where

$$A = \left.\frac{(s+2)}{(s+5)}\right|_{s\to 0} = \frac{2}{5}$$

$$B = \left.\frac{(s+2)}{s}\right|_{s\to -5} = \frac{3}{5}$$

Thus,

$$c(t) = \frac{2}{5} + \frac{3}{5}e^{-5t} \tag{4.2}$$

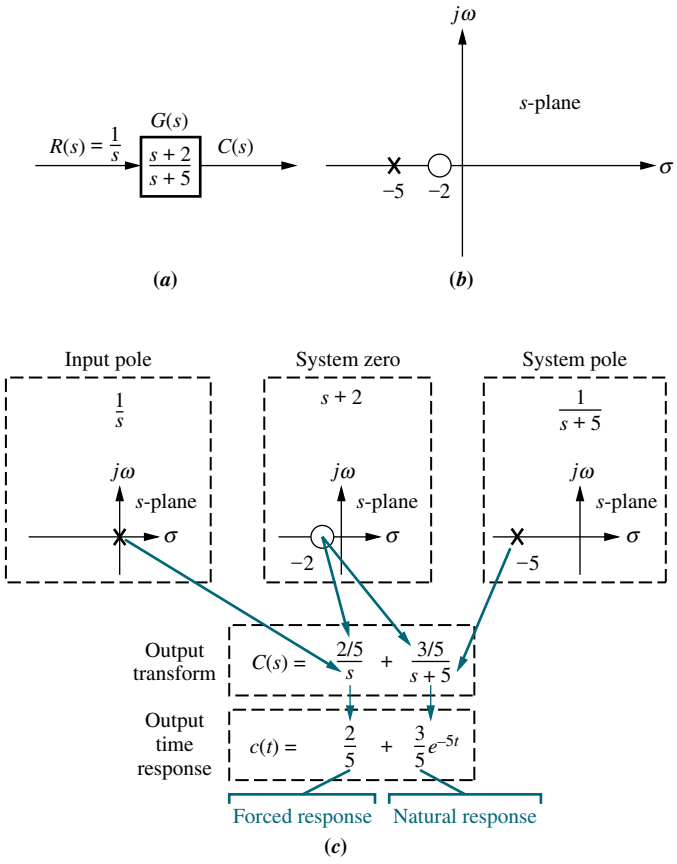


FIGURE 4.1 **a.** System showing input and output; **b.** pole-zero plot of the system; **c.** evolution of a system response. Follow blue arrows to see the evolution of the response component generated by the pole or zero.

From the development summarized in Figure 4.1(*c*), we draw the following conclusions:

1. A pole of the input function generates the form of the *forced response* (that is, the pole at the origin generated a step function at the output).
2. A pole of the transfer function generates the form of the *natural response* (that is, the pole at -5 generated e^{-5t}).
3. A pole on the real axis generates an *exponential* response of the form $e^{-\alpha t}$, where $-\alpha$ is the pole location on the real axis. Thus, the farther to the left a pole is on the negative real axis, the faster the exponential transient response will decay to zero (again, the pole at -5 generated e^{-5t}; see Figure 4.2 for the general case).
4. The zeros and poles generate the *amplitudes* for both the forced and natural responses (this can be seen from the calculation of A and B in Eq. (4.1)).

Let us now look at an example that demonstrates the technique of using poles to obtain the form of the system response. We will learn to write the form of the response by inspection. Each pole of the system transfer function that is on the real axis generates an exponential response that is a component of the natural response. The input pole generates the forced response.

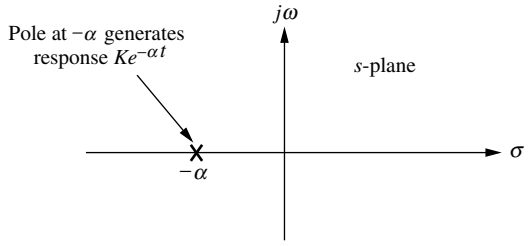


FIGURE 4.2 Effect of a real-axis pole upon transient response.

Example 4.1

Evaluating Response Using Poles

PROBLEM: Given the system of Figure 4.3, write the output, $c(t)$, in general terms. Specify the forced and natural parts of the solution.

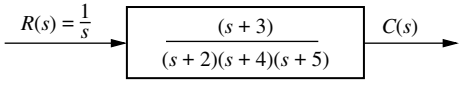


FIGURE 4.3 System for Example 4.1

SOLUTION: By inspection, each system pole generates an exponential as part of the natural response. The input's pole generates the forced response. Thus,

$$C(s) \equiv \underbrace{\frac{K_1}{s}}_{\text{Forced response}} + \underbrace{\frac{K_2}{s+2} + \frac{K_3}{s+4} + \frac{K_4}{s+5}}_{\text{Natural response}} \tag{4.3}$$

Taking the inverse Laplace transform, we get

$$c(t) \equiv \underbrace{K_1}_{\text{Forced response}} + \underbrace{K_{2e^{-2t}} + K_{3e^{-4t}} + K_{4e^{-5t}}}_{\text{Natural response}} \tag{4.4}$$

Skill-Assessment Exercise 4.1

PROBLEM: A system has a transfer function, $G(s) = \dfrac{10(s+4)(s+6)}{(s+1)(s+7)(s+8)(s+10)}$.

Write, by inspection, the output, $c(t)$, in general terms if the input is a unit step.

ANSWER: $c(t) \equiv A + Be^{-t} + Ce^{-7t} + De^{-8t} + Ee^{-10t}$

In this section, we learned that poles determine the nature of the time response: Poles of the input function determine the form of the forced response, and poles of the transfer function determine the form of the natural response. Zeros and poles of the input or transfer function contribute to the amplitudes of the component parts of the total response. Finally, poles on the real axis generate exponential responses.

4.3 First-Order Systems

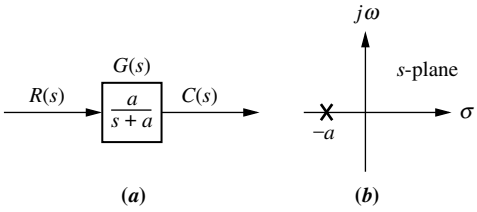


FIGURE 4.4 **a.** First-order system; **b.** pole plot

We now discuss first-order systems without zeros to define a performance specification for such a system. A first-order system without zeros can be described by the transfer function shown in Figure 4.4(*a*). If the input is a unit step, where $R(s) = 1/s$, the Laplace transform of the step response is $C(s)$, where

$$C(s) = R(s)G(s) = \frac{a}{s(s+a)} \tag{4.5}$$

Taking the inverse transform, the step response is given by

$$\boxed{c(t) = c_f(t) + c_n(t) = 1 - e^{-at}} \tag{4.6}$$

where the input pole at the origin generated the forced response $c_f(t) = 1$, and the system pole at $-a$, as shown in Figure 4.4(*b*), generated the natural response $c_n(t) = -e^{-at}$. Equation (4.6) is plotted in Figure 4.5.

Let us examine the significance of parameter a, the only parameter needed to describe the transient response. When $t = 1/a$,

$$e^{-at}|_{t=1/a} = e^{-1} = 0.37 \tag{4.7}$$

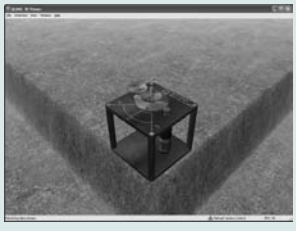
or

$$c(t)|_{t=1/a} = 1 - e^{-at}|_{t=1/a} = 1 - 0.37 = 0.63 \tag{4.8}$$

We now use Eqs. (4.6), (4.7), and (4.8) to define three transient response performance specifications.

Virtual Experiment 4.1
First-Order Open-Loop Systems

Put theory into practice and find a first-order transfer function representing the Quanser Rotary Servo. Then validate the model by simulating it in LabVIEW. Such a servo motor is used in mechatronic gadgets such as cameras.



Virtual experiments are found on WileyPLUS.

Time Constant

We call $1/a$ the *time constant* of the response. From Eq. (4.7), the time constant can be described as the time for e^{-at} to decay to 37% of its initial value. Alternately, from Eq. (4.8) the time constant is the time it takes for the step response to rise to 63% of its final value (see Figure 4.5).

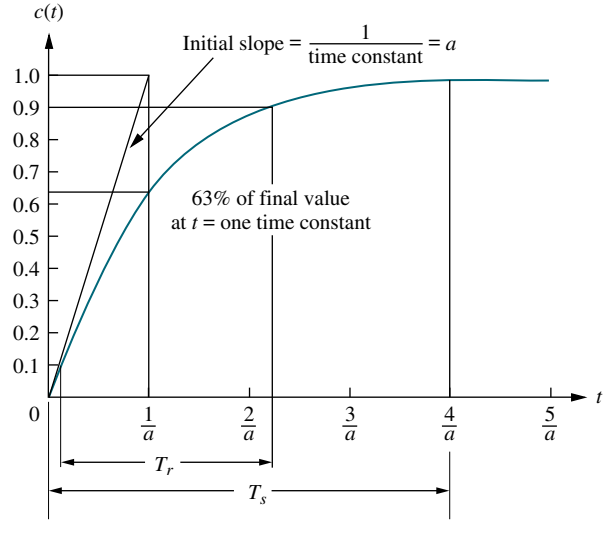


FIGURE 4.5 First-order system response to a unit step

The reciprocal of the time constant has the units (1/seconds), or frequency. Thus, we can call the parameter a the *exponential frequency*. Since the derivative of e^{-at} is $-a$ when $t = 0$, a is the initial rate of change of the exponential at $t = 0$. Thus, the time constant can be considered a transient response specification for a first-order system, since it is related to the speed at which the system responds to a step input.

The time constant can also be evaluated from the pole plot (see Figure 4.4(b)). Since the pole of the transfer function is at $-a$, we can say the pole is located at the *reciprocal* of the time constant, and the farther the pole from the imaginary axis, the faster the transient response.

Let us look at other transient response specifications, such as rise time, T_r, and settling time, T_s, as shown in Figure 4.5.

Rise Time, T_r

Rise time is defined as the time for the waveform to go from 0.1 to 0.9 of its final value. Rise time is found by solving Eq. (4.6) for the difference in time at $c(t) = 0.9$ and $c(t) = 0.1$. Hence,

$$T_r = \frac{2.31}{a} - \frac{0.11}{a} = \frac{2.2}{a} \tag{4.9}$$

Settling Time, T_s

Settling time is defined as the time for the response to reach, and stay within, 2% of its final value.[2] Letting $c(t) = 0.98$ in Eq. (4.6) and solving for time, t, we find the settling time to be

$$T_s = \frac{4}{a} \tag{4.10}$$

First-Order Transfer Functions via Testing

Often it is not possible or practical to obtain a system's transfer function analytically. Perhaps the system is closed, and the component parts are not easily identifiable. Since the transfer function is a representation of the system from input to output, the system's step response can lead to a representation even though the inner construction is not known. With a step input, we can measure the time constant and the steady-state value, from which the transfer function can be calculated.

Consider a simple first-order system, $G(s) = K/(s + a)$, whose step response is

$$C(s) = \frac{K}{s(s+a)} = \frac{K/a}{s} - \frac{K/a}{(s+a)} \tag{4.11}$$

If we can identify K and a from laboratory testing, we can obtain the transfer function of the system.

For example, assume the unit step response given in Figure 4.6. We determine that it has the first-order characteristics we have seen thus far, such as no overshoot and nonzero initial slope. From the response, we measure the time constant, that is, the time for the amplitude to reach 63% of its final value. Since the final value is

[2] Strictly speaking, this is the definition of the 2% *setting time*. Other percentages, for example 5%, also can be used. We will use *settling time* throughout the book to mean 2% settling time.

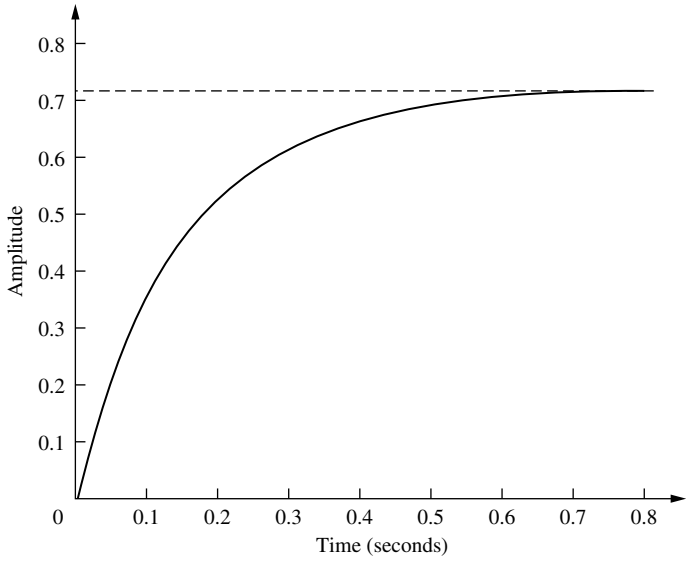


FIGURE 4.6 Laboratory results of a system step response test

about 0.72, the time constant is evaluated where the curve reaches $0.63 \times 0.72 = 0.45$, or about 0.13 second. Hence, $a = 1/0.13 = 7.7$.

To find K, we realize from Eq. (4.11) that the forced response reaches a steady-state value of $K/a = 0.72$. Substituting the value of a, we find $K = 5.54$. Thus, the transfer function for the system is $G(s) = 5.54/(s + 7.7)$. It is interesting to note that the response of Figure 4.6 was generated using the transfer function $G(s) = 5/(s + 7)$.

Skill-Assessment Exercise 4.2

PROBLEM: A system has a transfer function, $G(s) = \dfrac{50}{s + 50}$. Find the time constant, T_c, settling time, T_s, and rise time, T_r.

ANSWER: $T_c = 0.02\,\text{s}$, $T_s = 0.08\,\text{s}$, and $T_r = 0.044\,\text{s}$.

The complete solution is located at www.wiley.com/college/nise.

4.4 Second-Order Systems: Introduction

Let us now extend the concepts of poles and zeros and transient response to second-order systems. Compared to the simplicity of a first-order system, a second-order system exhibits a wide range of responses that must be analyzed and described. Whereas varying a first-order system's parameter simply changes the speed of the response, changes in the parameters of a second-order system can change the *form* of the response. For example, a second-order system can display characteristics much

like a first-order system, or, depending on component values, display damped or pure oscillations for its transient response.

To become familiar with the wide range of responses before formalizing our discussion in the next section, we take a look at numerical examples of the second-order system responses shown in Figure 4.7. All examples are derived from Figure 4.7(*a*), the general case, which has two finite poles and no zeros. The term in the numerator is simply a scale or input multiplying factor that can take on any value without affecting the form of the derived results. By assigning appropriate values to parameters *a* and *b,* we can show all possible second-order transient responses. The unit step response then can be found using $C(s) = R(s)G(s)$, where $R(s) = 1/s$, followed by a partial-fraction expansion and the inverse Laplace transform. Details are left as an end-of-chapter problem, for which you may want to review Section 2.2.

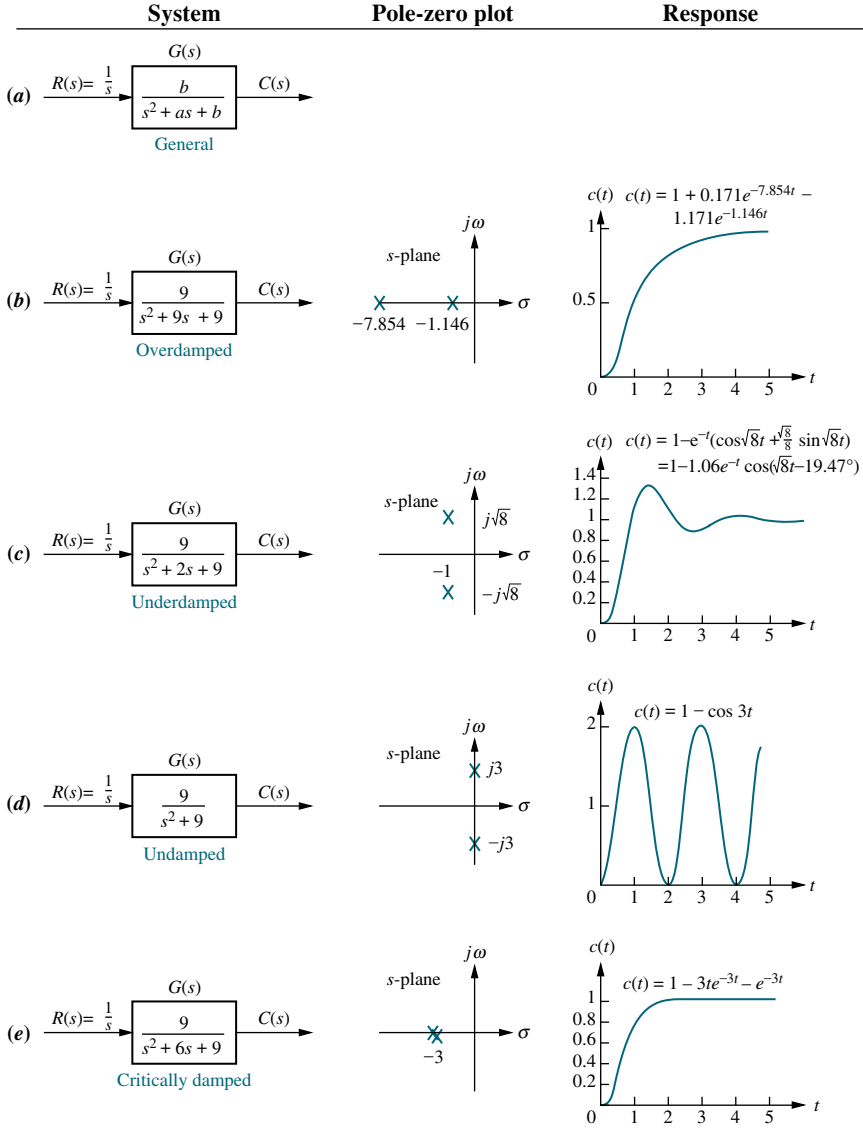


FIGURE 4.7 Second-order systems, pole plots, and step responses

We now explain each response and show how we can use the poles to determine the nature of the response without going through the procedure of a partial-fraction expansion followed by the inverse Laplace transform.

Overdamped Response, Figure 4.7(*b*)

For this response,

$$C(s) = \frac{9}{s(s^2 + 9s + 9)} = \frac{9}{s(s + 7.854)(s + 1.146)} \tag{4.12}$$

This function has a pole at the origin that comes from the unit step input and two real poles that come from the system. The input pole at the origin generates the constant forced response; each of the two system poles on the real axis generates an exponential natural response whose exponential frequency is equal to the pole location. Hence, the output initially could have been written as $c(t) = K_1 + K_2e^{-7.854t} + K_3e^{-1.146t}$. This response, shown in Figure 4.7(*b*), is called *overdamped*.[3] We see that the poles tell us the form of the response without the tedious calculation of the inverse Laplace transform.

Underdamped Response, Figure 4.7 (*c*)

For this response,

$$C(s) = \frac{9}{s(s^2 + 2s + 9)} \tag{4.13}$$

This function has a pole at the origin that comes from the unit step input and two complex poles that come from the system. We now compare the response of the second-order system to the poles that generated it. First we will compare the pole location to the time function, and then we will compare the pole location to the plot. From Figure 4.7(*c*), the poles that generate the natural response are at $s = -1 \pm j\sqrt{8}$. Comparing these values to $c(t)$ in the same figure, we see that the real part of the pole matches the exponential decay frequency of the sinusoid's amplitude, while the imaginary part of the pole matches the frequency of the sinusoidal oscillation.

Let us now compare the pole location to the plot. Figure 4.8 shows a general, damped sinusoidal response for a second-order system. The transient response consists of an exponentially decaying amplitude generated by the real part of the system pole times a sinusoidal waveform generated by the imaginary part of the system pole. The time constant of the exponential decay is equal to the reciprocal of the real part of the system pole. The value of the imaginary part is the actual frequency of the sinusoid, as depicted in Figure 4.8. This sinusoidal frequency is given the name *damped frequency of oscillation*, ω_d. Finally, the steady-state response (unit step) was generated by the input pole located at the origin. We call the type of response shown in Figure 4.8 an *underdamped response,* one which approaches a steady-state value via a transient response that is a damped oscillation.

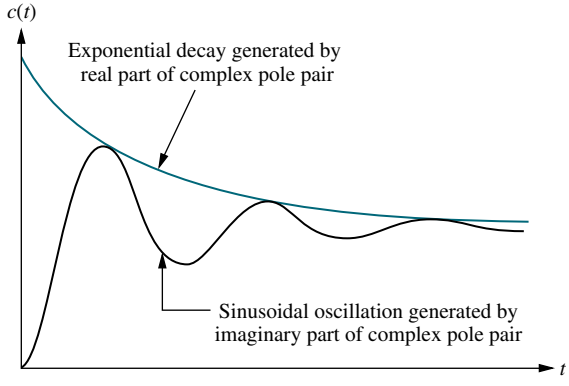


FIGURE 4.8 Second-order step response components generated by complex poles

The following example demonstrates how a knowledge of the relationship between the pole location and the transient response can lead rapidly to the response form without calculating the inverse Laplace transform.

[3] So named because *overdamped* refers to a large amount of energy absorption in the system, which inhibits the transient response from overshooting and oscillating about the steady-state value for a step input. As the energy absorption is reduced, an overdamped system will become underdamped and exhibit overshoot.

Example 4.2

Form of Underdamped Response Using Poles

PROBLEM: By inspection, write the form of the step response of the system in Figure 4.9.

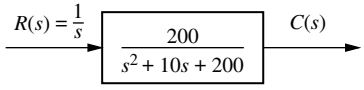


FIGURE 4.9 System for Example 4.2

SOLUTION: First we determine that the form of the forced response is a step. Next we find the form of the natural response. Factoring the denominator of the transfer function in Figure 4.9, we find the poles to be $s = -5 \pm j13.23$. The real part, -5, is the exponential frequency for the damping. It is also the reciprocal of the time constant of the decay of the oscillations. The imaginary part, 13.23, is the radian frequency for the sinusoidal oscillations. Using our previous discussion and Figure 4.7(c) as a guide, we obtain $c(t) = K_1 + e^{-5t}(K_2 \cos 13.23t + K_3 \sin 13.23t) = K_1 + K_4 e^{-5t}(\cos 13.23t - \phi)$, where $\phi = \tan^{-1} K_3/K_2$, $K_4 = \sqrt{K_2^2 + K_3^2}$, and $c(t)$ is a constant plus an exponentially damped sinusoid.

We will revisit the second-order underdamped response in Sections 4.5 and 4.6, where we generalize the discussion and derive some results that relate the pole position to other parameters of the response.

Undamped Response, Figure 4.7(*d*)

For this response,

$$C(s) = \frac{9}{s(s^2 + 9)} \tag{4.14}$$

This function has a pole at the origin that comes from the unit step input and two imaginary poles that come from the system. The input pole at the origin generates the constant forced response, and the two system poles on the imaginary axis at $\pm j3$ generate a sinusoidal natural response whose frequency is equal to the location of the imaginary poles. Hence, the output can be estimated as $c(t) = K_1 + K_4 \cos(3t - \phi)$. This type of response, shown in Figure 4.7(d), is called *undamped*. Note that the absence of a real part in the pole pair corresponds to an exponential that does not decay. Mathematically, the exponential is $e^{-0t} = 1$.

Critically Damped Response, Figure 4.7 (*e*)

For this response,

$$C(s) = \frac{9}{s(s^2 + 6s + 9)} = \frac{9}{s(s + 3)^2} \tag{4.15}$$

This function has a pole at the origin that comes from the unit step input and two multiple real poles that come from the system. The input pole at the origin generates the constant forced response, and the two poles on the real axis at -3 generate a natural response consisting of an exponential and an exponential multiplied by time, where the exponential frequency is equal to the location of the real poles. Hence, the output can be estimated as $c(t) = K_1 + K_2 e^{-3t} + K_3 t e^{-3t}$. This type of response, shown in Figure 4.7(e), is called *critically damped*. Critically damped responses are the fastest possible without the overshoot that is characteristic of the underdamped response.

We now summarize our observations. In this section we defined the following natural responses and found their characteristics:

1. *Overdamped responses*

Poles: Two real at $-\sigma_1,\ -\sigma_2$

Natural response: Two exponentials with time constants equal to the reciprocal of the pole locations, or

$$c(t) = K_1 e^{-\sigma_1 t} + K_2 e^{-\sigma_2 t}$$

2. *Underdamped responses*

Poles: Two complex at $-\sigma_d \pm j\omega_d$

Natural response: Damped sinusoid with an exponential envelope whose time constant is equal to the reciprocal of the pole's real part. The radian frequency of the sinusoid, the damped frequency of oscillation, is equal to the imaginary part of the poles, or

$$c(t) = Ae^{-\sigma_d t}\cos(\omega_d t - \phi)$$

3. *Undamped responses*

Poles: Two imaginary at $\pm j\omega_1$

Natural response: Undamped sinusoid with radian frequency equal to the imaginary part of the poles, or

$$c(t) = \text{A}\cos(\omega_1 t - \phi)$$

4. *Critically damped responses*

Poles: Two real at $-\sigma_1$

Natural response: One term is an exponential whose time constant is equal to the reciprocal of the pole location. Another term is the product of time, t, and an exponential with time constant equal to the reciprocal of the pole location, or

$$c(t) = K_1 e^{-\sigma_1 t} + K_2 t e^{-\sigma_1 t}$$

The step responses for the four cases of damping discussed in this section are superimposed in Figure 4.10. Notice that the critically damped case is the division

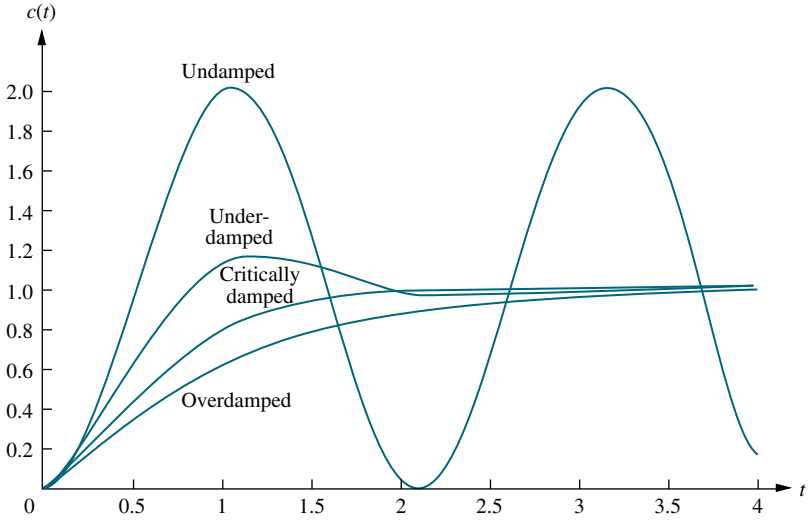


FIGURE 4.10 Step responses for second-order system damping cases

between the overdamped cases and the underdamped cases and is the fastest response without overshoot.

Skill-Assessment Exercise 4.3

PROBLEM: For each of the following transfer functions, write, by inspection, the general form of the step response:

WileyPLUS
WPCS
Control Solutions

a. $G(s) = \dfrac{400}{s^2 + 12s + 400}$

b. $G(s) = \dfrac{900}{s^2 + 90s + 900}$

c. $G(s) = \dfrac{225}{s^2 + 30s + 225}$

d. $G(s) = \dfrac{625}{s^2 + 625}$

ANSWERS:

a. $c(t) = A + Be^{-6t}\cos(19.08t + \phi)$

b. $c(t) = A + Be^{-78.54t} + Ce^{-11.46t}$

c. $c(t) = A + Be^{-15t} + Cte^{-15t}$

d. $c(t) = A + B\cos(25t + \phi)$

The complete solution is located at www.wiley.com/college/nise.

In the next section, we will formalize and generalize our discussion of second-order responses and define two specifications used for the analysis and design of second-order systems. In Section 4.6, we will focus on the *underdamped* case and derive some specifications unique to this response that we will use later for analysis and design.

4.5 The General Second-Order System

Now that we have become familiar with second-order systems and their responses, we generalize the discussion and establish quantitative specifications defined in such a way that the response of a second-order system can be described to a designer without the need for sketching the response. In this section, we define two physically meaningful specifications for second-order systems. These quantities can be used to describe the characteristics of the second-order transient response just as time constants describe the first-order system response. The two quantities are called natural frequency and damping ratio. Let us formally define them.

Natural Frequency, ω_n

The *natural frequency* of a second-order system is the frequency of oscillation of the system without damping. For example, the frequency of oscillation of a series *RLC* circuit with the resistance shorted would be the natural frequency.

Damping Ratio, ζ

Before we state our next definition, some explanation is in order. We have already seen that a second-order system's underdamped step response is characterized by damped oscillations. Our definition is derived from the need to quantitatively describe this damped oscillation regardless of the time scale. Thus, a system whose transient response goes through three cycles in a millisecond before reaching the steady state would have the same measure as a system that went through three cycles in a millennium before reaching the steady state. For example, the underdamped curve in Figure 4.10 has an associated measure that defines its shape. This measure remains the same even if we change the time base from seconds to microseconds or to millennia.

A viable definition for this quantity is one that compares the exponential decay frequency of the envelope to the natural frequency. This ratio is constant regardless of the time scale of the response. Also, the reciprocal, which is proportional to the ratio of the natural period to the exponential time constant, remains the same regardless of the time base.

We define the *damping ratio*, ζ, to be

$$\zeta = \frac{\text{Exponential decay frequency}}{\text{Natural frequency (rad/second)}} = \frac{1}{2\pi}\,\frac{\text{Natural period (seconds)}}{\text{Exponential time constant}}$$

Let us now revise our description of the second-order system to reflect the new definitions. The general second-order system shown in Figure 4.7(a) can be transformed to show the quantities ζ and ω_n. Consider the general system

$$G(s) = \frac{b}{s^2 + as + b} \tag{4.16}$$

Without damping, the poles would be on the $j\omega$-axis, and the response would be an undamped sinusoid. For the poles to be purely imaginary, $a = 0$. Hence,

$$G(s) = \frac{b}{s^2 + b} \tag{4.17}$$

By definition, the natural frequency, ω_n, is the frequency of oscillation of this system. Since the poles of this system are on the $j\omega$-axis at $\pm j\sqrt{b}$,

$$\omega_n = \sqrt{b} \tag{4.18}$$

Hence,

$$b = \omega_n^2 \tag{4.19}$$

Now what is the term a in Eq. (4.16)? Assuming an underdamped system, the complex poles have a real part, σ, equal to $-a/2$. The magnitude of this value is then the exponential decay frequency described in Section 4.4. Hence,

$$\zeta = \frac{\text{Exponential decay frequency}}{\text{Natural frequency (rad/second)}} = \frac{|\sigma|}{\omega_n} = \frac{a/2}{\omega_n} \tag{4.20}$$

from which

$$a = 2\zeta\omega_n \tag{4.21}$$

Our general second-order transfer function finally looks like this:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.22}$$

In the following example we find numerical values for ζ and ω_n by matching the transfer function to Eq. (4.22).

Example 4.3

Finding ζ and ω_n For a Second-Order System

PROBLEM: Given the transfer function of Eq. (4.23), find ζ and ω_n.

$$G(s) = \frac{36}{s^2 + 4.2s + 36} \tag{4.23}$$

SOLUTION: Comparing Eq. (4.23) to (4.22), $\omega_n^2 = 36$, from which $\omega_n = 6$. Also, $2\zeta\omega_n = 4.2$. Substituting the value of ω_n, $\zeta = 0.35$.

Now that we have defined ζ and ω_n, let us relate these quantities to the pole location. Solving for the poles of the transfer function in Eq. (4.22) yields

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \tag{4.24}$$

From Eq. (4.24) we see that the various cases of second-order response are a function of ζ; they are summarized in Figure 4.11.[4]

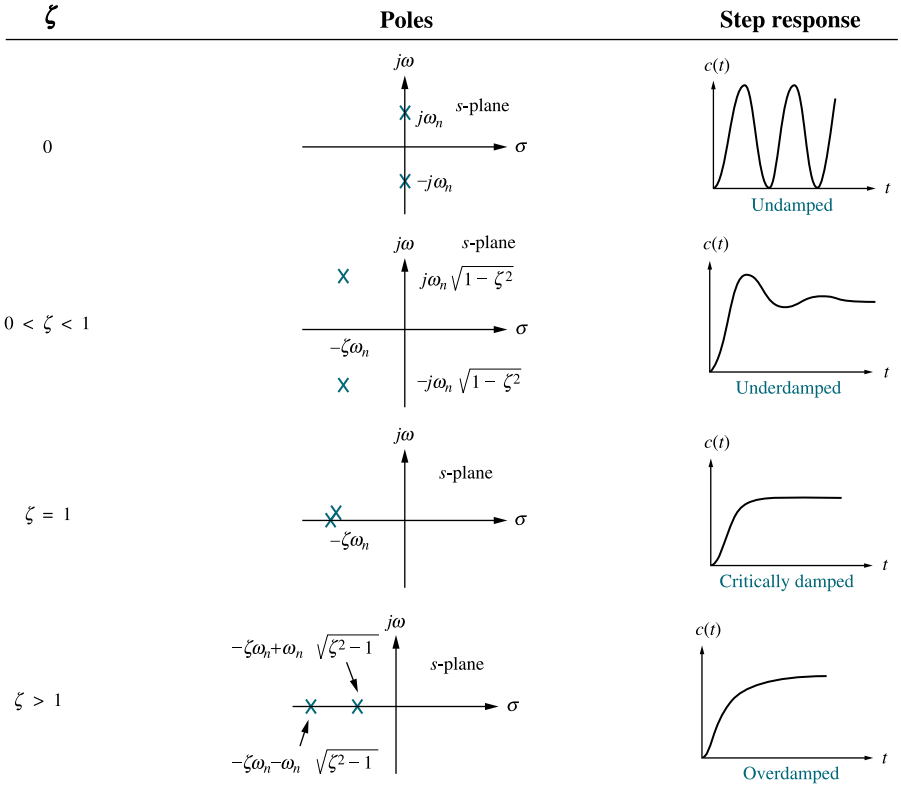


FIGURE 4.11 Second-order response as a function of damping ratio

[4] The student should verify Figure 4.11 as an exercise.

In the following example we find the numerical value of ζ and determine the nature of the transient response.

Example 4.4

Characterizing Response from the Value of ζ

PROBLEM: For each of the systems shown in Figure 4.12, find the value of ζ and report the kind of response expected.

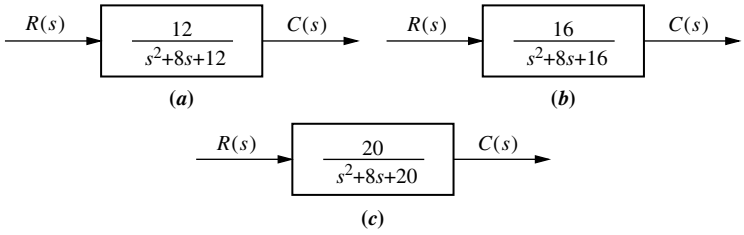


FIGURE 4.12 Systems for Example 4.4

SOLUTION: First match the form of these systems to the forms shown in Eqs. (4.16) and (4.22). Since $a = 2\zeta\omega_n$ and $\omega_n = \sqrt{b}$,

$$\zeta = \frac{a}{2\sqrt{b}} \tag{4.25}$$

Using the values of a and b from each of the systems of Figure 4.12, we find $\zeta = 1.155$ for system (a), which is thus overdamped, since $\zeta > 1$; $\zeta = 1$ for system (b), which is thus critically damped; and $\zeta = 0.894$ for system (c), which is thus underdamped, since $\zeta < 1$.

Skill-Assessment Exercise 4.4

PROBLEM: For each of the transfer functions in Skill-Assessment Exercise 4.3, do the following: (1) Find the values of ζ and ω_n; (2) characterize the nature of the response.

ANSWERS:

a. $\zeta = 0.3$, $\omega_n = 20$; system is underdamped
b. $\zeta = 1.5$, $\omega_n = 30$; system is overdamped
c. $\zeta = 1$, $\omega_n = 15$; system is critically damped
d. $\zeta = 0$, $\omega_n = 25$; system is undamped

The complete solution is located at www.wiley.com/college/nise.

This section defined two specifications, or parameters, of second-order systems: natural frequency, ω_n, and damping ratio, ζ. We saw that the nature of the response obtained was related to the value of ζ. Variations of damping ratio alone yield the complete range of overdamped, critically damped, underdamped, and undamped responses.

4.6 Underdamped Second-Order Systems

Now that we have generalized the second-order transfer function in terms of ζ and ω_n, let us analyze the step response of an *underdamped* second-order system. Not only will this response be found in terms of ζ and ω_n, but more specifications indigenous to the underdamped case will be defined. The underdamped second-order system, a common model for physical problems, displays unique behavior that must be itemized; a detailed description of the underdamped response is necessary for both analysis and design. Our first objective is to define transient specifications associated with underdamped responses. Next we relate these specifications to the pole location, drawing an association between pole location and the form of the underdamped second-order response. Finally, we tie the pole location to system parameters, thus closing the loop: Desired response generates required system components.

Let us begin by finding the step response for the general second-order system of Eq. (4.22). The transform of the response, $C(s)$, is the transform of the input times the transfer function, or

$$C(s) = \frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{K_1}{s} + \frac{K_2 s + K_3}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.26}$$

where it is assumed that $\zeta < 1$ (the underdamped case). Expanding by partial fractions, using the methods described in Section 2.2, Case 3, yields

$$C(s) = \frac{1}{s} - \frac{(s + \zeta\omega_n) + \frac{\zeta}{\sqrt{1-\zeta^2}}\omega_n\sqrt{1-\zeta^2}}{(s + \zeta\omega_n)^2 + \omega_n^2(1-\zeta^2)} \tag{4.27}$$

Taking the inverse Laplace transform, which is left as an exercise for the student, produces

$$\begin{aligned} c(t) &= 1 - e^{-\zeta\omega_n t}\left(\cos\omega_n\sqrt{1-\zeta^2}t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_n\sqrt{1-\zeta^2}t\right) \\ &= 1 - \frac{1}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_n t}\cos(\omega_n\sqrt{1-\zeta^2}t - \phi) \end{aligned} \tag{4.28}$$

where $\phi = \tan^{-1}(\zeta/\sqrt{1-\zeta^2})$.

A plot of this response appears in Figure 4.13 for various values of ζ, plotted along a time axis normalized to the natural frequency. We now see the relationship between the value of ζ and the type of response obtained: The lower the value of ζ, the more oscillatory the response. The natural frequency is a time-axis scale factor and does not affect the nature of the response other than to scale it in time.

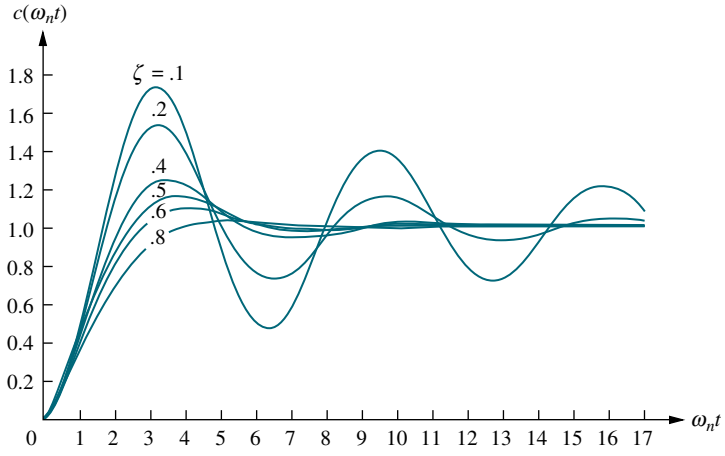


FIGURE 4.13 Second-order underdamped responses for damping ratio values

We have defined two parameters associated with second-order systems, ζ and ω_n. Other parameters associated with the underdamped response are rise time, peak time, percent overshoot, and settling time. These specifications are defined as follows (see also Figure 4.14):

1. *Rise time,* T_r. The time required for the waveform to go from 0.1 of the final value to 0.9 of the final value.
2. *Peak time,* T_P. The time required to reach the first, or maximum, peak.
3. *Percent overshoot,* $\%OS$. The amount that the waveform overshoots the steady-state, or final, value at the peak time, expressed as a percentage of the steady-state value.
4. *Settling time,* T_s. The time required for the transient's damped oscillations to reach and stay within $\pm 2\%$ of the steady-state value.

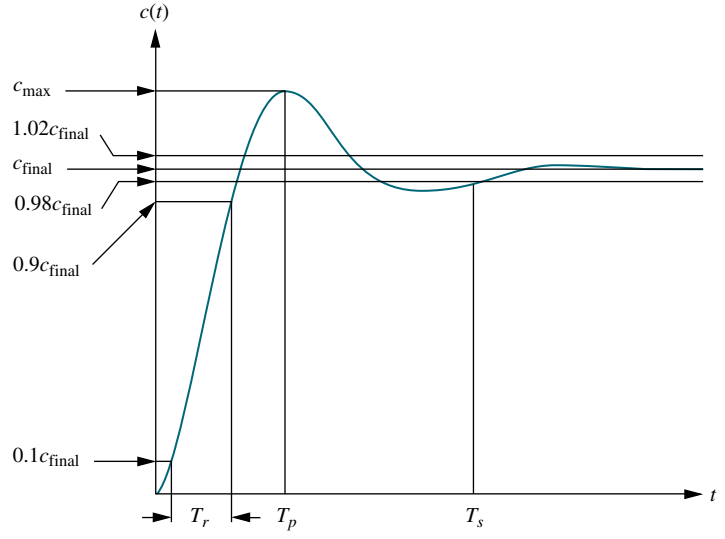


FIGURE 4.14 Second-order underdamped response specifications

Notice that the definitions for settling time and rise time are basically the same as the definitions for the first-order response. All definitions are also valid for systems of order higher than 2, although analytical expressions for these parameters cannot be found unless the response of the higher-order system can be approximated as a second-order system, which we do in Sections 4.7 and 4.8.

Rise time, peak time, and settling time yield information about the speed of the transient response. This information can help a designer determine if the speed and the nature of the response do or do not degrade the performance of the system. For example, the speed of an entire computer system depends on the time it takes for a hard drive head to reach steady state and read data; passenger comfort depends in part on the suspension system of a car and the number of oscillations it goes through after hitting a bump.

We now evaluate T_p, $\%OS$, and T_s as functions of ζ and ω_n. Later in this chapter we relate these specifications to the location of the system poles. A precise analytical expression for rise time cannot be obtained; thus, we present a plot and a table showing the relationship between ζ and rise time.

Evaluation of T_p

T_p is found by differentiating $c(t)$ in Eq. (4.28) and finding the first zero crossing after $t = 0$. This task is simplified by "differentiating" in the frequency domain by using Item 7 of Table 2.2. Assuming zero initial conditions and using Eq. (4.26), we get

$$\mathscr{L}[\dot{c}(t)] = sC(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.29}$$

Completing squares in the denominator, we have

$$\mathscr{L}[\dot{c}(t)] = \frac{\omega_n^2}{(s + \zeta\omega_n)^2 + \omega_n^2(1 - \zeta^2)} = \frac{\frac{\omega_n}{\sqrt{1-\zeta^2}}\omega_n\sqrt{1 - \zeta^2}}{(s + \zeta\omega_n)^2 + \omega_n^2(1 - \zeta^2)} \tag{4.30}$$

Therefore,

$$\dot{c}(t) = \frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin\omega_n\sqrt{1 - \zeta^2}t \tag{4.31}$$

Setting the derivative equal to zero yields

$$\omega_n\sqrt{1 - \zeta^2}t = n\pi \tag{4.32}$$

or

$$t = \frac{n\pi}{\omega_n\sqrt{1 - \zeta^2}} \tag{4.33}$$

Each value of n yields the time for local maxima or minima. Letting $n = 0$ yields $t = 0$, the first point on the curve in Figure 4.14 that has zero slope. The first peak, which occurs at the peak time, T_p, is found by letting $n = 1$ in Eq. (4.33):

$$\boxed{T_p = \frac{\pi}{\omega_n\sqrt{1 - \zeta^2}}} \tag{4.34}$$

Evaluation of %OS

From Figure 4.14 the percent overshoot, $\%OS$, is given by

$$\%OS = \frac{c_{\max} - c_{\text{final}}}{c_{\text{final}}} \times 100 \tag{4.35}$$

The term $c_{\max}$ is found by evaluating $c(t)$ at the peak time, $c(T_p)$. Using Eq. (4.34) for T_p and substituting into Eq. (4.28) yields

$$\begin{aligned} c_{\max} = c(T_p) &= 1 - e^{-(\zeta\pi/\sqrt{1-\zeta^2})}\left(\cos\pi + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\pi\right) \\ &= 1 + e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \end{aligned} \tag{4.36}$$

For the unit step used for Eq. (4.28),

$$c_{\text{final}} = 1 \tag{4.37}$$

Substituting Eqs. (4.36) and (4.37) into Eq. (4.35), we finally obtain

$$\boxed{\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100} \tag{4.38}$$

Notice that the percent overshoot is a function only of the damping ratio, ζ.

Whereas Eq. (4.38) allows one to find $\%OS$ given ζ, the inverse of the equation allows one to solve for ζ given $\%OS$. The inverse is given by

$$\boxed{\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}}} \tag{4.39}$$

The derivation of Eq. (4.39) is left as an exercise for the student. Equation (4.38) (or, equivalently, (4.39)) is plotted in Figure 4.15.

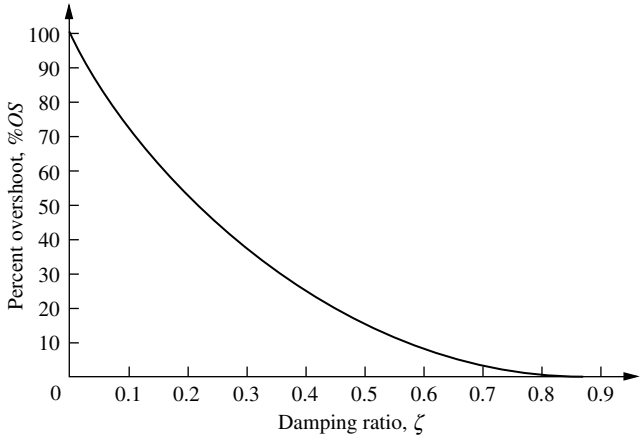


FIGURE 4.15 Percent overshoot versus damping ratio

Evaluation of T_s

In order to find the settling time, we must find the time for which $c(t)$ in Eq. (4.28) reaches and stays within $\pm 2\%$ of the steady-state value, c_{final}. Using our definition, the settling time is the time it takes for the amplitude of the decaying sinusoid in Eq. (4.28) to reach 0.02, or

$$e^{-\zeta\omega_n t}\frac{1}{\sqrt{1-\zeta^2}} = 0.02 \tag{4.40}$$

This equation is a conservative estimate, since we are assuming that $\cos(\omega_n\sqrt{1-\zeta^2}t - \phi) = 1$ at the settling time. Solving Eq. (4.40) for t, the settling time is

$$T_s = \frac{-\ln(0.02\sqrt{1-\zeta^2})}{\zeta\omega_n} \tag{4.41}$$

You can verify that the numerator of Eq. (4.41) varies from 3.91 to 4.74 as ζ varies from 0 to 0.9. Let us agree on an approximation for the settling time that will be used for all values of ζ; let it be

$$\boxed{T_s = \frac{4}{\zeta\omega_n}} \tag{4.42}$$

Evaluation of T_r

A precise analytical relationship between rise time and damping ratio, ζ, cannot be found. However, using a computer and Eq. (4.28), the rise time can be found. We first designate $\omega_n t$ as the normalized time variable and select a value for ζ. Using the computer, we solve for the values of $\omega_n t$ that yield $c(t) = 0.9$ and $c(t) = 0.1$. Subtracting the two values of $\omega_n t$ yields the normalized rise time, $\omega_n T_r$, for that value of ζ. Continuing in like fashion with other values of ζ, we obtain the results plotted in Figure 4.16.[5] Let us look at an example.

Damping ratio	Normalized rise time
0.1	1.104
0.2	1.203
0.3	1.321
0.4	1.463
0.5	1.638
0.6	1.854
0.7	2.126
0.8	2.467
0.9	2.883

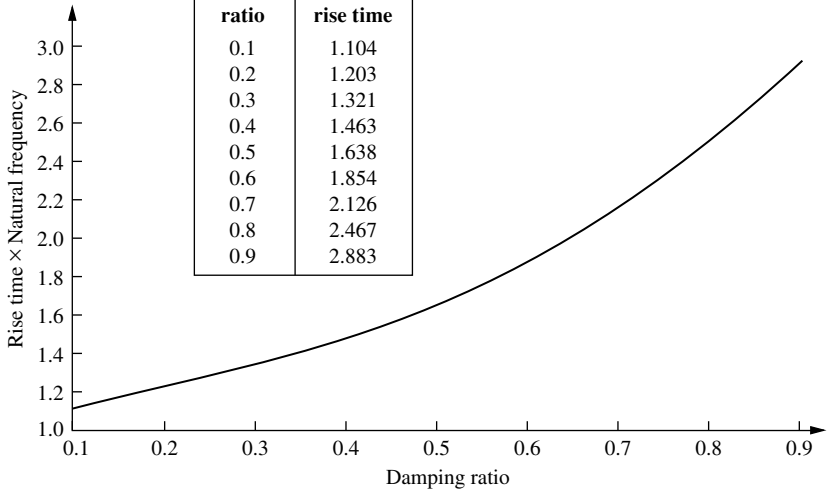


FIGURE 4.16 Normalized rise time versus damping ratio for a second-order underdamped response

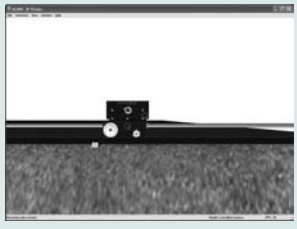
[5] Figure 4.16 can be approximated by the following polynomials: $\omega_n T_r = 1.76\zeta^3 - 0.417\zeta^2 + 1.039\zeta + 1$ (maximum error less than $\frac{1}{2}\%$ for $0 < \zeta < 0.9$), and $\zeta = 0.115(\omega_n T_r)^3 - 0.883(\omega_n T_r)^2 + 2.504(\omega_n T_r) - 1.738$ (maximum error less than 5% for $0.1 < \zeta < 0.9$). The polynomials were obtained using MATLAB's **polyfit** function.

Example 4.5

Finding T_p, %OS, T_s, and T_r from a Transfer Function

Virtual Experiment 4.2 Second-Order System Response

Put theory into practice studying the effect that natural frequency and damping ratio have on controlling the speed response of the Quanser Linear Servo in LabVIEW. This concept is applicable to automobile cruise controls or speed controls of subways or trucks.



Virtual experiments are found on WileyPLUS.

PROBLEM: Given the transfer function

$$G(s) = \frac{100}{s^2 + 15s + 100} \tag{4.43}$$

find T_p, $\%OS$, T_s, and T_r.

SOLUTION: ω_n and ζ are calculated as 10 and 0.75, respectively. Now substitute ζ and ω_n into Eqs. (4.34), (4.38), and (4.42) and find, respectively, that $T_p = 0.475$ second, $\%OS = 2.838$, and $T_s = 0.533$ second. Using the table in Figure 4.16, the normalized rise time is approximately 2.3 seconds. Dividing by ω_n yields $T_r = 0.23$ second. This problem demonstrates that we can find T_p, $\%OS$, T_s, and T_r without the tedious task of taking an inverse Laplace transform, plotting the output response, and taking measurements from the plot.

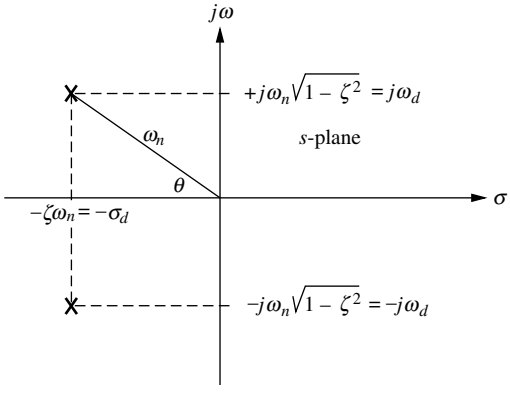


FIGURE 4.17 Pole plot for an underdamped second-order system

We now have expressions that relate peak time, percent overshoot, and settling time to the natural frequency and the damping ratio. Now let us relate these quantities to the location of the poles that generate these characteristics.

The pole plot for a general, underdamped second-order system, previously shown in Figure 4.11, is reproduced and expanded in Figure 4.17 for focus. We see from the Pythagorean theorem that the radial distance from the origin to the pole is the natural frequency, ω_n, and the $\cos\theta = \zeta$.

Now, comparing Eqs. (4.34) and (4.42) with the pole location, we evaluate peak time and settling time in terms of the pole location. Thus,

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \frac{\pi}{\omega_d} \tag{4.44}$$

$$T_s = \frac{4}{\zeta\omega_n} = \frac{\pi}{\sigma_d} \tag{4.45}$$

where ω_d is the imaginary part of the pole and is called the *damped frequency of oscillation*, and σ_d is the magnitude of the real part of the pole and is the *exponential damping frequency.*

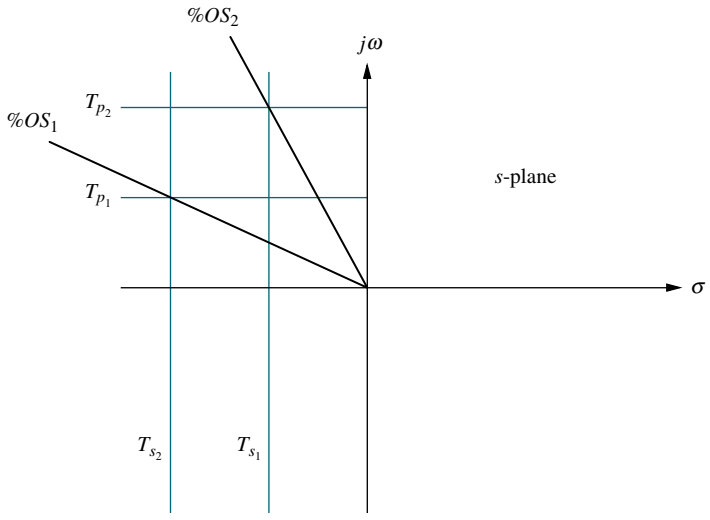


FIGURE 4.18 Lines of constant peak time, T_p, settling time, T_s, and percent overshoot, $\%OS$. Note: $T_{s_2} < T_{s_1}$; $T_{p2} < T_{p1}$; $\%OS_1 < \%OS_2$.

Equation (4.44) shows that T_p is inversely proportional to the imaginary part of the pole. Since horizontal lines on the s-plane are lines of constant imaginary value, they are also lines of constant peak time. Similarly, Eq. (4.45) tells us that settling time is inversely proportional to the real part of the pole. Since vertical lines on the s-plane are lines of constant real value, they are also lines of constant settling time. Finally, since $\zeta = \cos\theta$, radial lines are lines of constant ζ. Since percent overshoot is only a function of ζ, radial lines are thus lines of constant percent overshoot, $\%OS$. These concepts are depicted in Figure 4.18, where lines of constant T_p, T_s, and $\%OS$ are labeled on the s-plane.

At this point, we can understand the significance of Figure 4.18 by examining the actual step response of comparative systems. Depicted in Figure 4.19(a) are the step responses as the poles are moved in a vertical direction, keeping the real part the same. As the poles move in a vertical direction, the frequency increases, but the envelope remains the same since the real part of the pole is not changing. The figure shows a constant exponential envelope, even though the sinusoidal response is changing frequency. Since all curves fit under the same exponential decay curve, the settling time is virtually the same for all waveforms. Note that as overshoot increases, the rise time decreases.

Let us move the poles to the right or left. Since the imaginary part is now constant, movement of the poles yields the responses of Figure 4.19(b). Here the frequency is constant over the range of variation of the real part. As the poles move to the left, the response damps out more rapidly, while the frequency remains the same. Notice that the peak time is the same for all waveforms because the imaginary part remains the same.

Moving the poles along a constant radial line yields the responses shown in Figure 4.19(c). Here the percent overshoot remains the same. Notice also that the responses look exactly alike, except for their speed. The farther the poles are from the origin, the more rapid the response.

We conclude this section with some examples that demonstrate the relationship between the pole location and the specifications of the second-order underdamped response. The first example covers analysis. The second example is a simple design problem consisting of a physical system whose component values we want to design to meet a transient response specification.

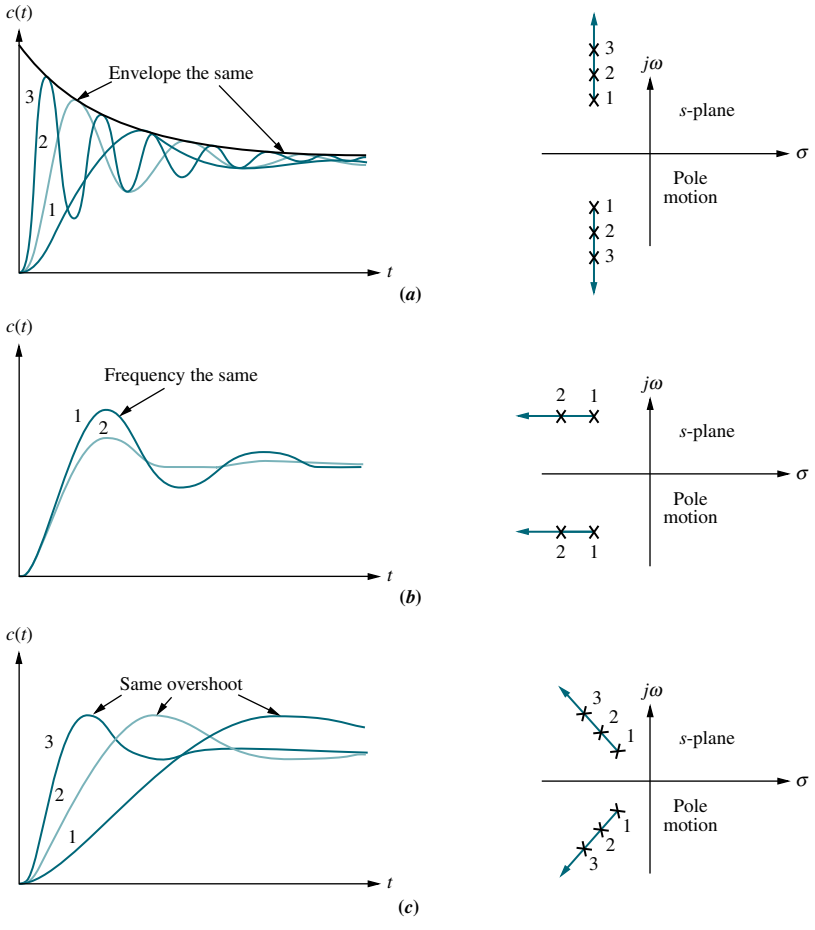


FIGURE 4.19 Step responses of second-order underdamped systems as poles move: **a.** with constant real part; **b.** with constant imaginary part; **c.** with constant damping ratio

Example 4.6

Finding T_p, $\%OS$, and T_s from Pole Location

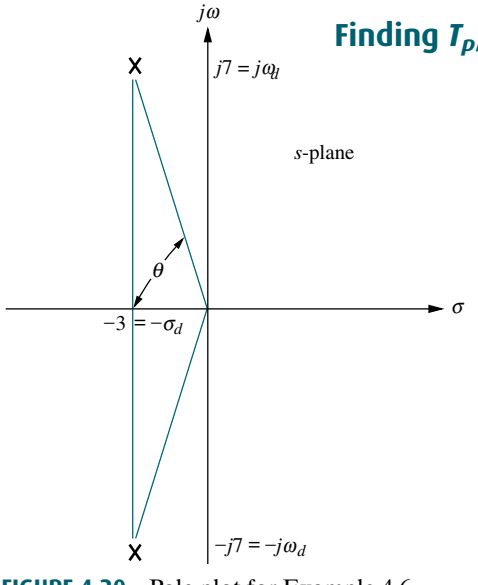


FIGURE 4.20 Pole plot for Example 4.6

PROBLEM: Given the pole plot shown in Figure 4.20, find ζ, ω_n, T_p, $\%OS$, and T_s.

SOLUTION: The damping ratio is given by $\zeta = \cos\theta = \cos[\arctan(7/3)] = 0.394$. The natural frequency, ω_n, is the radial distance from the origin to the pole, or $\omega_n = \sqrt{7^2 + 3^2} = 7.616$. The peak time is

$$T_p = \frac{\pi}{\omega_d} = \frac{\pi}{7} = 0.449 \text{ second} \tag{4.46}$$

The percent overshoot is

$$\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100 = 26\% \tag{4.47}$$

The approximate settling time is

$$T_s = \frac{4}{\sigma_d} = \frac{4}{3} = 1.333 \text{ seconds} \tag{4.48}$$

MATLAB
ML

Students who are using MATLAB should now run ch4p1 in Appendix B. You will learn how to generate a second-order polynomial from two complex poles as well as extract and use the coefficients of the polynomial to calculate T_p, %OS, and T_s. This exercise uses MATLAB to solve the problem in Example 4.6.

Example 4.7

Transient Response Through Component Design

Design
D

PROBLEM: Given the system shown in Figure 4.21, find J and D to yield 20% overshoot and a settling time of 2 seconds for a step input of torque $T(t)$.

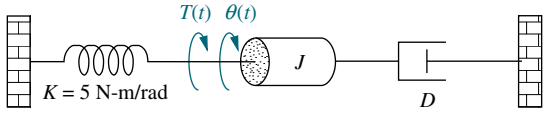


FIGURE 4.21 Rotational mechanical system for Example 4.7

SOLUTION: First, the transfer function for the system is

$$G(s) = \frac{1/J}{s^2 + \frac{D}{J}s + \frac{K}{J}} \tag{4.49}$$

From the transfer function,

$$\omega_n = \sqrt{\frac{K}{J}} \tag{4.50}$$

and

$$2\zeta\omega_n = \frac{D}{J} \tag{4.51}$$

But, from the problem statement,

$$T_s = 2 = \frac{4}{\zeta\omega_n} \tag{4.52}$$

or $\zeta\omega_n = 2$. Hence,

$$2\zeta\omega_n = 4 = \frac{D}{J} \tag{4.53}$$

Also, from Eqs. (4.50) and (4.52),

$$\zeta = \frac{4}{2\omega_n} = 2\sqrt{\frac{J}{K}} \tag{4.54}$$

From Eq. (4.39), a 20% overshoot implies $\zeta = 0.456$. Therefore, from Eq. (4.54),

$$\zeta = 2\sqrt{\frac{J}{K}} = 0.456 \tag{4.55}$$

Hence,

$$\frac{J}{K} = 0.052 \tag{4.56}$$

From the problem statement, $K = 5$ N-m/rad. Combining this value with Eqs. (4.53) and (4.56), $D = 1.04$ N-m-s/rad, and $J = 0.26$ kg-m^2.

Second-Order Transfer Functions via Testing

Just as we obtained the transfer function of a first-order system experimentally, we can do the same for a system that exhibits a typical underdamped second-order response. Again, we can measure the laboratory response curve for percent overshoot and settling time, from which we can find the poles and hence the denominator. The numerator can be found, as in the first-order system, from a knowledge of the measured and expected steady-state values. A problem at the end of the chapter illustrates the estimation of a second-order transfer function from the step response.

Skill-Assessment Exercise 4.5

TryIt 4.1

Use the following MATLAB statements to calculate the answers to Skill-Assessment Exercise 4.5. Ellipses mean code continues on next line.

```
numg=361;
deng=[1 16 361];
omegan=sqrt(deng(3)...
 /deng(1))
zeta=(deng(2)/deng(1))...
 /(2*omegan)
Ts=4/(zeta*omegan)
Tp=pi/(omegan*sqrt...
 (1-zeta^2))
pos=100*exp(-zeta*...
 pi/sqrt(1-zeta^2))
Tr=(1.768*zeta^3 -...
 0.417*zeta^2+1.039*...
 zeta+1)/omegan
```

WileyPLUS WPCS Control Solutions

PROBLEM: Find ζ, ω_n, T_s, T_p, T_r, and $\%OS$ for a system whose transfer function is $G(s) = \dfrac{361}{s^2 + 16s + 361}$.

ANSWERS:

$\zeta = 0.421$, $\omega_n = 19$, $T_s = 0.5$ s, $T_p = 0.182$ s, $T_r = 0.079$ s, and $\%OS = 23.3\%$.

The complete solution is located at www.wiley.com/college/nise.

Now that we have analyzed systems with two poles, how does the addition of another pole affect the response? We answer this question in the next section.

4.7 System Response with Additional Poles

In the last section, we analyzed systems with one or two poles. It must be emphasized that the formulas describing percent overshoot, settling time, and peak time were derived only for a system with two complex poles and no zeros. If a system such as that shown in Figure 4.22 has more than two poles or has zeros, we cannot use the formulas to calculate the performance specifications that we derived. However, under certain conditions, a system with more than two poles or with zeros can be



FIGURE 4.22 Robot follows input commands from a human trainer

approximated as a second-order system that has just two complex *dominant poles*. Once we justify this approximation, the formulas for percent overshoot, settling time, and peak time can be applied to these higher-order systems by using the location of the dominant poles. In this section, we investigate the effect of an additional pole on the second-order response. In the next section, we analyze the effect of adding a zero to a two-pole system.

Let us now look at the conditions that would have to exist in order to approximate the behavior of a three-pole system as that of a two-pole system. Consider a three-pole system with complex poles and a third pole on the real axis. Assuming that the complex poles are at $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$ and the real pole is at $-\alpha_r$, the step response of the system can be determined from a partial-fraction expansion. Thus, the output transform is

$$C(s) = \frac{A}{s} + \frac{B(s+\zeta\omega_n) + C\omega_d}{(s+\zeta\omega_n)^2 + \omega_d^2} + \frac{D}{s+\alpha_r} \tag{4.57}$$

or, in the time domain,

$$c(t) = Au(t) + e^{-\zeta\omega_n t}(B\cos\omega_d t + C\sin\omega_d t) + De^{-\alpha_r t} \tag{4.58}$$

The component parts of $c(t)$ are shown in Figure 4.23 for three cases of α_r. For Case I, $\alpha_r = \alpha_{r_1}$ and is not much larger than $\zeta\omega_n$; for Case II, $\alpha_r = \alpha_{r_2}$ and is much larger than $\zeta\omega_n$; and for Case III, $\alpha_r = \infty$.

Let us direct our attention to Eq. (4.58) and Figure 4.23. If $\alpha_r \gg \zeta\omega_n$ (Case II), the pure exponential will die out much more rapidly than the second-order underdamped step response. If the pure exponential term decays to an insignificant value at the time of the first overshoot, such parameters as percent overshoot, settling time, and peak time will be generated by the second-order underdamped step response component. Thus, the total response will approach that of a pure second-order system (Case III).

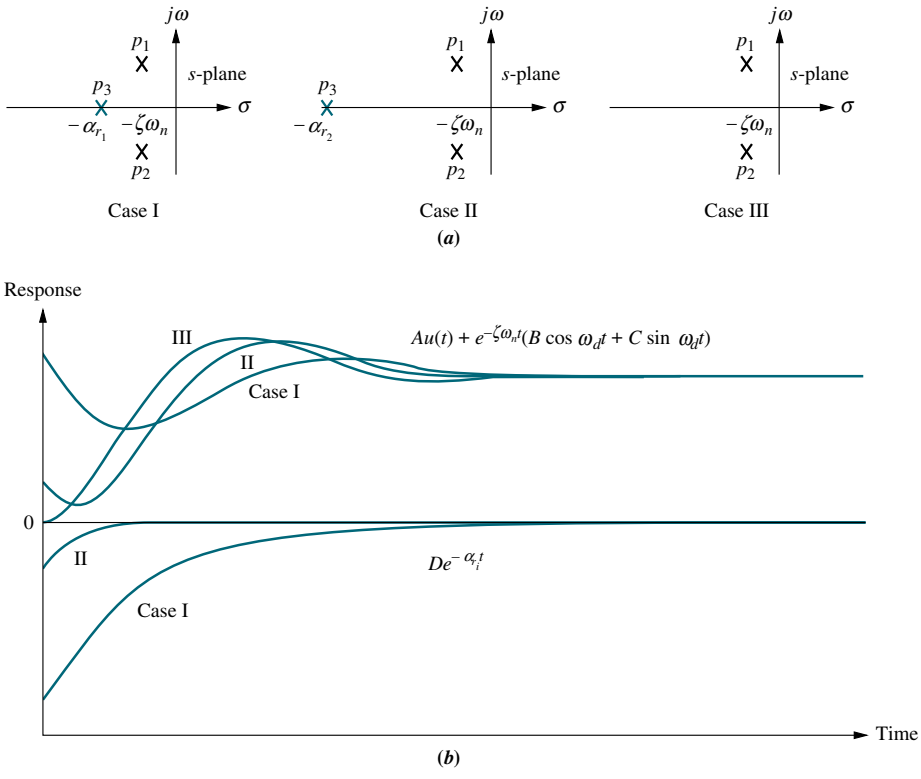


FIGURE 4.23 Component responses of a three-pole system: **a.** pole plot; **b.** component responses: Nondominant pole is near dominant second-order pair (Case I), far from the pair (Case II), and at infinity (Case III)

If α_r is not much greater than $\zeta\omega_n$ (Case I), the real pole's transient response will not decay to insignificance at the peak time or settling time generated by the second-order pair. In this case, the exponential decay is significant, and the system cannot be represented as a second-order system.

The next question is, How much farther from the dominant poles does the third pole have to be for its effect on the second-order response to be negligible? The answer of course depends on the accuracy for which you are looking. However, this book assumes that the exponential decay is negligible after five time constants. Thus, if the real pole is five times farther to the left than the dominant poles, we assume that the system is represented by its dominant second-order pair of poles.

What about the magnitude of the exponential decay? Can it be so large that its contribution at the peak time is not negligible? We can show, through a partial-fraction expansion, that the residue of the third pole, in a three-pole system with dominant second-order poles and no zeros, will actually decrease in magnitude as the third pole is moved farther into the left half-plane. Assume a step response, $C(s)$, of a three-pole system:

$$C(s) = \frac{bc}{s(s^2 + as + b)(s + c)} = \frac{A}{s} + \frac{Bs + C}{s^2 + as + b} + \frac{D}{s + c} \tag{4.59}$$

where we assume that the nondominant pole is located at $-c$ on the real axis and that the steady-state response approaches unity. Evaluating the constants in the numerator of each term,

$$A = 1; \qquad B = \frac{ca - c^2}{c^2 + b - ca} \tag{4.60a}$$

$$C = \frac{ca^2 - c^2a - bc}{c^2 + b - ca}; \quad D = \frac{-b}{c^2 + b - ca} \tag{4.60b}$$

As the nondominant pole approaches ∞, or $c \to \infty$,

$$A = 1;\ B = -1;\ C = -a;\ D = 0 \tag{4.61}$$

Thus, for this example, D, the residue of the nondominant pole and its response, becomes zero as the nondominant pole approaches infinity.

The designer can also choose to forgo extensive residue analysis, since all system designs should be simulated to determine final acceptance. In this case, the control systems engineer can use the "five times" rule of thumb as a necessary but not sufficient condition to increase the confidence in the second-order approximation during design, but then simulate the completed design.

Let us look at an example that compares the responses of two different three-pole systems with that of a second-order system.

Example 4.8

Comparing Responses of Three-Pole Systems

PROBLEM: Find the step response of each of the transfer functions shown in Eqs. (4.62) through (4.64) and compare them.

$$T_1(s) = \frac{24.542}{s^2 + 4s + 24.542} \tag{4.62}$$

$$T_2(s) = \frac{245.42}{(s + 10)(s^2 + 4s + 24.542)} \tag{4.63}$$

$$T_3(s) = \frac{73.626}{(s + 3)(s^2 + 4s + 24.542)} \tag{4.64}$$

SOLUTION: The step response, $C_i(s)$, for the transfer function, $T_i(s)$, can be found by multiplying the transfer function by $1/s$, a step input, and using partial-fraction expansion followed by the inverse Laplace transform to find the response, $c_i(t)$. With the details left as an exercise for the student, the results are

$$c_1(t) = 1 - 1.09e^{-2t}\cos(4.532t - 23.8^\circ) \tag{4.65}$$

$$c_2(t) = 1 - 0.29e^{-10t} - 1.189e^{-2t}\cos(4.532t - 53.34^\circ) \tag{4.66}$$

$$c_3(t) = 1 - 1.14e^{-3t} + 0.707e^{-2t}\cos(4.532t + 78.63^\circ) \tag{4.67}$$

The three responses are plotted in Figure 4.24. Notice that $c_2(t)$, with its third pole at -10 and farthest from the dominant poles, is the better approximation of $c_1(t)$,

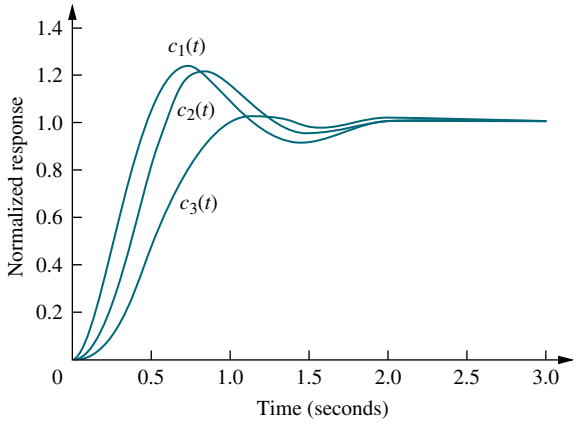


FIGURE 4.24 Step responses of system $T_1(s)$, system $T_2(s)$, and system $T_3(s)$

the pure second-order system response; $c_3(t)$, with a third pole close to the dominant poles, yields the most error.

MATLAB
ML

Students who are using MATLAB should now run ch4p2 in Appendix B. You will learn how to generate a step response for a transfer function and how to plot the response directly or collect the points for future use. The example shows how to collect the points and then use them to create a multiple plot, title the graph, and label the axes and curves to produce the graph in Figure 4.24 to solve Example 4.8.

Simulink
SL

System responses can alternately be obtained using Simulink. Simulink is a software package that is integrated with MATLAB to provide a graphical user interface (GUI) for defining systems and generating responses. The reader is encouraged to study Appendix C, which contains a tutorial on Simulink as well as some examples. One of the illustrative examples, Example C.1, solves Example 4.8 using Simulink.

Gui Tool
GUIT

Another method to obtain systems responses is through the use of MATLAB's LTI Viewer. An advantage of the LTI Viewer is that it displays the values of settling time, peak time, rise time, maximum response, and the final value on the step response plot. The reader is encouraged to study Appendix E at www.wiley.com/college/nise, which contains a tutorial on the LTI Viewer as well as some examples. Example E.1 solves Example 4.8 using the LTI Viewer.

Skill-Assessment Exercise 4.6

TryIt 4.2

Use the following MATLAB and Control System Toolbox statements to investigate the effect of the additional pole in Skill-Assessment Exercise 4.6(a). Move the higher-order pole originally at −15 to other values by changing "a" in the code.

```
a=15
numga=100*a;
denga=conv([1 a],...
 [1 4 100]);
Ta=tf(numga,denga);
numg=100;
deng=[1 4 100];
T=tf (numg,deng);
step(Ta,'.',T,'-')
```

PROBLEM: Determine the validity of a second-order approximation for each of these two transfer functions:

a. $G(s) = \dfrac{700}{(s+15)(s^2+4s+100)}$

b. $G(s) = \dfrac{360}{(s+4)(s^2+2s+90)}$

ANSWERS:

a. The second-order approximation is valid.

b. The second-order approximation is not valid.

The complete solution is located at www.wiley.com/college/nise.

4.8 System Response With Zeros

Now that we have seen the effect of an additional pole, let us add a zero to the second-order system. In Section 4.2, we saw that the zeros of a response affect the residue, or amplitude, of a response component but do not affect the nature of the response—exponential, damped sinusoid, and so on. In this section, we add a real-axis zero to a two-pole system. The zero will be added first in the left half-plane and then in the right half-plane and its effects noted and analyzed. We conclude the section by talking about pole-zero cancellation.

Starting with a two-pole system with poles at $(-1 \pm j2.828)$, we consecutively add zeros at -3, -5, and -10. The results, normalized to the steady-state value, are plotted in Figure 4.25. We can see that the closer the zero is to the dominant poles, the greater its effect on the transient response. As the zero moves away from the dominant poles, the response approaches that of the two-pole system. This analysis can be reasoned via the partial-fraction expansion. If we assume a group of poles and a zero far from the poles, the residue of each pole will be affected the same by the zero. Hence, the relative amplitudes remain appreciably the same. For example, assume the partial-fraction expansion shown in Eq. (4.68):

$$\begin{aligned} T(s) = \frac{(s+a)}{(s+b)(s+c)} &= \frac{A}{s+b} + \frac{B}{s+c} \\ &= \frac{(-b+a)/(-b+c)}{s+b} + \frac{(-c+a)/(-c+b)}{s+c} \end{aligned} \tag{4.68}$$

If the zero is far from the poles, then a is large compared to b and c, and

$$T(s) \approx a\left[\frac{1/(-b+c)}{s+b} + \frac{1/(-c+b)}{s+c}\right] = \frac{a}{(s+b)(s+c)} \tag{4.69}$$

Hence, the zero looks like a simple gain factor and does not change the relative amplitudes of the components of the response.

Another way to look at the effect of a zero, which is more general, is as follows (*Franklin, 1991*): Let $C(s)$ be the response of a system, $T(s)$, with unity in the

TryIt 4.3

Use the following MATLAB and Control System Toolbox statements to generate Figure 4.25.

```
deng=[1 2 9];
Ta=tf([1 3]*9/3,deng);
Tb=tf([1 5]*9/5,deng);
Tc=tf([1 10]*9/10,deng);
T=tf(9,deng);
step(T,Ta,Tb,Tc)
text(0.5,0.6,'no zero')
text(0.4,0.7,...
 'zero at -10')
text(0.35,0.8,...
 'zero at -5')
text(0.3,0.9,'zero at -3')
```

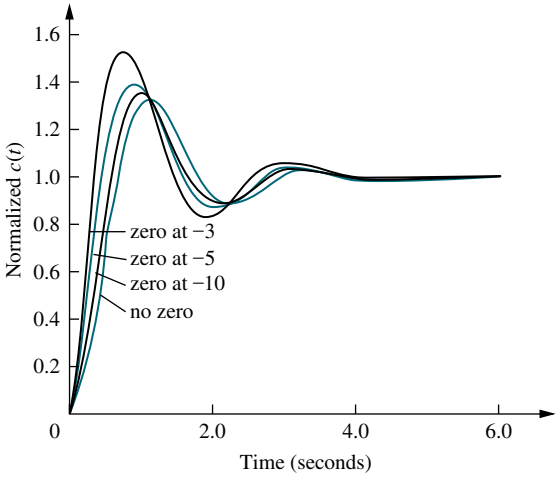


FIGURE 4.25 Effect of adding a zero to a two-pole system

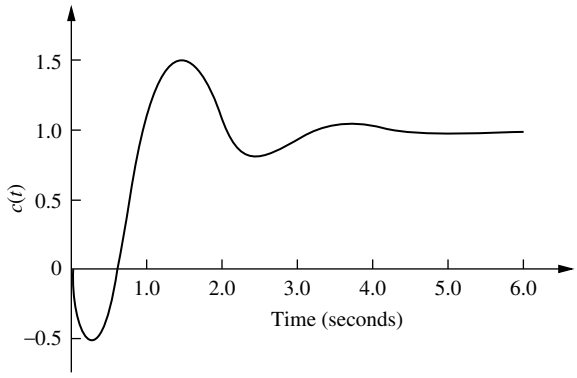


FIGURE 4.26 Step response of a nonminimum-phase system

numerator. If we add a zero to the transfer function, yielding $(s + a)T(s)$, the Laplace transform of the response will be

$$(s + a)C(s) = sC(s) + aC(s) \tag{4.70}$$

Thus, the response of a system with a zero consists of two parts: the derivative of the original response and a scaled version of the original response. If a, the negative of the zero, is very large, the Laplace transform of the response is approximately $aC(s)$, or a scaled version of the original response. If a is not very large, the response has an additional component consisting of the derivative of the original response. As a becomes smaller, the derivative term contributes more to the response and has a greater effect. For step responses, the derivative is typically positive at the start of a step response. Thus, for small values of a, we can expect more overshoot in second-order systems because the derivative term will be additive around the first overshoot. This reasoning is borne out by Figure 4.25.

An interesting phenomenon occurs if a is negative, placing the zero in the right half-plane. From Eq. (4.70) we see that the derivative term, which is typically positive initially, will be of opposite sign from the scaled response term. Thus, if the derivative term, $sC(s)$, is larger than the scaled response, $aC(s)$, the response will initially follow the derivative in the opposite direction from the scaled response. The result for a second-order system is shown in Figure 4.26, where the sign of the input was reversed to yield a positive steady-state value. Notice that the response begins to turn toward the negative direction even though the final value is positive. A system that exhibits this phenomenon is known as a *nonminimum-phase* system. If a motorcycle or airplane was a nonminimum-phase system, it would initially veer left when commanded to steer right.

Let us now look at an example of an electrical nonminimum-phase network.

Example 4.9

Transfer Function of a Nonminimum-Phase System

PROBLEM:

a. Find the transfer function, $V_o(s)/V_i(s)$ for the operational amplifier circuit shown in Figure 4.27.

b. If $R_1 = R_2$, this circuit is known as an all-pass filter, since it passes sine waves of a wide range of frequencies without attenuating or amplifying their magnitude (*Dorf, 1993*). We will learn more about frequency response in Chapter 10. For now, let $R_1 = R_2,\ R_3C = 1/10$, and find the step response of the filter. Show that component parts of the response can be identified with those in Eq. (4.70).

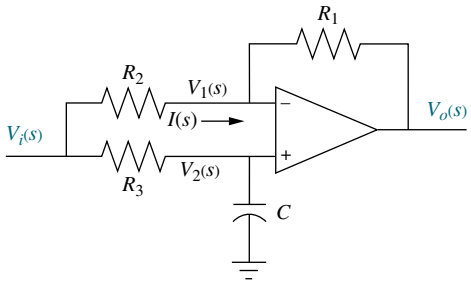


FIGURE 4.27 Nonminimum-phase electric circuit (Reprinted with permission of John Wiley & Sons, Inc.)

SOLUTION:

a. Remembering from Chapter 2 that the operational amplifier has a high input impedance, the current, $I(s)$, through R_1 and R_2, is the same and is equal to

$$I(s) = \frac{V_i(s) - V_o(s)}{R_1 + R_2} \tag{4.71}$$

Also,

$$V_o(s) = A(V_2(s) - V_1(s)) \tag{4.72}$$

But

$$V_1(s) = I(s)R_1 + V_o(s) \tag{4.73}$$

Substituting Eq. (4.71) into (4.73),

$$V_1(s) = \frac{1}{R_1 + R_2}(R_1V_i(s) + R_2V_0(s)) \tag{4.74}$$

Using voltage division,

$$V_2(s) = V_i(s)\frac{1/Cs}{R_3 + \dfrac{1}{Cs}} \tag{4.75}$$

Substituting Eqs. (4.74) and (4.75) into Eq. (4.72) and simplifying yields

$$\frac{V_o(s)}{V_i(s)} = \frac{A(R_2 - R_1R_3Cs)}{(R_3Cs + 1)(R_1 + R_2(1 + A))} \tag{4.76}$$

Since the operational amplifier has a large gain, A, let A approach infinity. Thus, after simplification

$$\frac{V_o(s)}{V_i(s)} = \frac{R_2 - R_1R_3Cs}{R_2R_3Cs + R_2} = -\frac{R_1}{R_2}\frac{\left(s - \dfrac{R_2}{R_1R_3C}\right)}{\left(s + \dfrac{1}{R_3C}\right)} \tag{4.77}$$

b. Letting $R_1 = R_2$ and $R_3C = 1/10$,

$$\frac{V_o(s)}{V_i(s)} = \frac{\left(s - \dfrac{1}{R_3C}\right)}{\left(s + \dfrac{1}{R_3C}\right)} = -\frac{(s - 10)}{(s + 10)} \tag{4.78}$$

For a step input, we evaluate the response as suggested by Eq. (4.70):

$$C(s) = -\frac{(s-10)}{s(s+10)} = -\frac{1}{s+10} + 10\,\frac{1}{s(s+10)} = sC_o(s) - 10C_o(s) \quad (4.79)$$

where

$$C_o(s) = -\frac{1}{s(s+10)} \quad (4.80)$$

is the Laplace transform of the response without a zero. Expanding Eq. (4.79) into partial fractions,

$$C(s) = -\frac{1}{s+10} + 10\frac{1}{s(s+10)} = -\frac{1}{s+10} + \frac{1}{s} - \frac{1}{s+10} = \frac{1}{s} - \frac{2}{s+10} \quad (4.81)$$

or the response with a zero is

$$c(t) = -e^{-10t} + 1 - e^{-10t} = 1 - 2e^{-10t} \quad (4.82)$$

Also, from Eq. (4.80),

$$C_o(s) = -\frac{1/10}{s} + \frac{1/10}{s+10} \quad (4.83)$$

or the response without a zero is

$$c_o(t) = -\frac{1}{10} + \frac{1}{10}e^{-10t} \quad (4.84)$$

The normalized responses are plotted in Figure 4.28. Notice the immediate reversal of the nonminimum-phase response, $c(t)$.

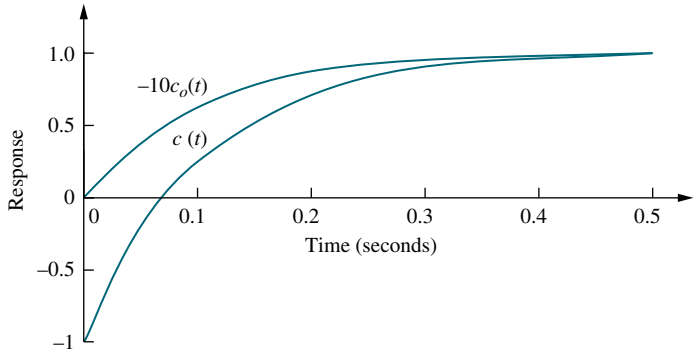


FIGURE 4.28 Step response of the nonminimum-phase network of Figure 4.27 ($c(t)$) and normalized step response of an equivalent network without the zero ($-10c_o(t)$)

We conclude this section by talking about pole-zero cancellation and its effect on our ability to make second-order approximations to a system. Assume a three-pole system with a zero as shown in Eq. (4.85). If the pole term, $(s+p_3)$, and the zero term, $(s+z)$, cancel out, we are left with

$$T(s) = \frac{K\cancel{(s+z)}}{\cancel{(s+p_3)}\,(s^2+as+b)} \quad (4.85)$$

as a second-order transfer function. From another perspective, if the zero at $-z$ is very close to the pole at $-p_3$, then a partial-fraction expansion of Eq. (4.85) will show that the residue of the exponential decay is much smaller than the amplitude of the second-order response. Let us look at an example.

Example 4.10

Evaluating Pole-Zero Cancellation Using Residues

PROBLEM: For each of the response functions in Eqs. (4.86) and (4.87), determine whether there is cancellation between the zero and the pole closest to the zero. For any function for which pole-zero cancellation is valid, find the approximate response.

$$C_1(s) = \frac{26.25(s+4)}{s(s+3.5)(s+5)(s+6)} \tag{4.86}$$

$$C_2(s) = \frac{26.25(s+4)}{s(s+4.01)(s+5)(s+6)} \tag{4.87}$$

SOLUTION: The partial-fraction expansion of Eq. (4.86) is

$$C_1(s) = \frac{1}{s} - \frac{3.5}{s+5} + \frac{3.5}{s+6} - \frac{1}{s+3.5} \tag{4.88}$$

The residue of the pole at -3.5, which is closest to the zero at -4, is equal to 1 and is not negligible compared to the other residues. Thus, a second-order step response approximation cannot be made for $C_1(s)$. The partial-fraction expansion for $C_2(s)$ is

$$C_2(s) = \frac{0.87}{s} - \frac{5.3}{s+5} + \frac{4.4}{s+6} + \frac{0.033}{s+4.01} \tag{4.89}$$

The residue of the pole at -4.01, which is closest to the zero at -4, is equal to 0.033, about two orders of magnitude below any of the other residues. Hence, we make a second-order approximation by neglecting the response generated by the pole at -4.01:

$$C_2(s) \approx \frac{0.87}{s} - \frac{5.3}{s+5} + \frac{4.4}{s+6} \tag{4.90}$$

and the response $c_2(t)$ is approximately

$$c_2(t) \approx 0.87 - 5.3e^{-5t} + 4.4e^{-6t} \tag{4.91}$$

TryIt 4.4

Use the following MATLAB and Symbolic Math Toolbox statements to evaluate the effect of higher-order poles by finding the component parts of the time response of $c_1(t)$ and $c_2(t)$ in Example 4.10.

```
syms s
C1=26.25*(s+4)/...
  (s*(s+3.5)*...
  (s+5)*(s+6));
C2=26.25*(s+4)/...
  (s*(s+4.01)*...
  (s+5)*(s+6));
c1=ilaplace(C1);
c1=vpa(c1,3);
'c1'
pretty(c1)
c2=ilaplace(C2);
c2=vpa(c2,3);
'c2'
pretty(c2);
```

Skill-Assessment Exercise 4.7

PROBLEM: Determine the validity of a second-order step-response approximation for each transfer function shown below.

WileyPLUS WPCS Control Solutions

a. $G(s) = \dfrac{185.71(s+7)}{(s+6.5)(s+10)(s+20)}$

b. $G(s) = \dfrac{197.14(s+7)}{(s+6.9)(s+10)(s+20)}$

ANSWERS:

a. A second-order approximation is not valid.

b. A second-order approximation is valid.

The complete solution is located at www.wiley.com/college/nise.

In this section, we have examined the effects of additional transfer function poles and zeros upon the response. In the next section we add nonlinearities of the type discussed in Section 2.10 and see what effects they have on system response.

4.9 Effects of Nonlinearities Upon Time Response

In this section, we qualitatively examine the effects of nonlinearities upon the time response of physical systems. In the following examples, we insert nonlinearities, such as saturation, dead zone, and backlash, as shown in Figure 2.46, into a system to show the effects of these nonlinearities upon the linear responses.

The responses were obtained using Simulink, a simulation software package that is integrated with MATLAB to provide a graphical user interface (GUI). Readers who would like to learn how to use Simulink to generate nonlinear responses should consult the Simulink tutorial in Appendix C. Simulink block diagrams are included with all responses that follow.

Let us assume the motor and load from the Antenna Control Case Study of Chapter 2 and look at the load angular velocity, $\omega_o(s)$, where $\omega_o(s) = 0.1\,s\theta_m(s) = 0.2083\,\mathrm{E}_a(s)/(s + 1.71)$ from Eq. (2.208). If we drive the motor with a step input

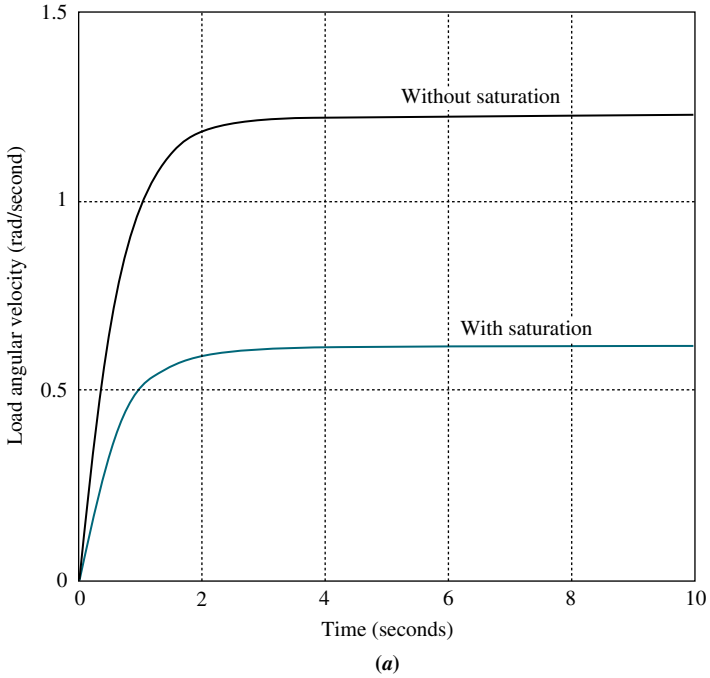
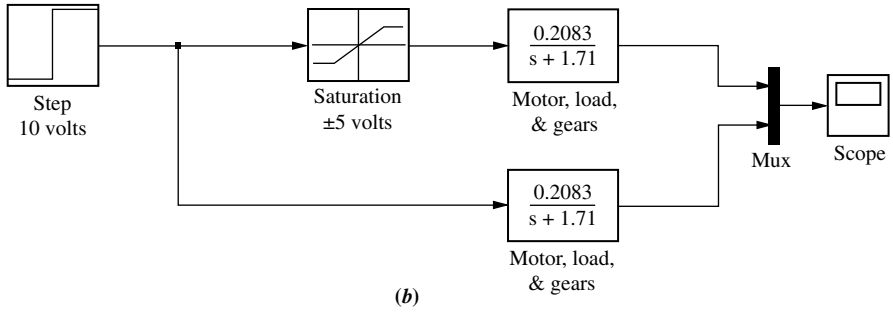


FIGURE 4.29 **a.** Effect of amplifier saturation on load angular velocity response; (*figure continues*)



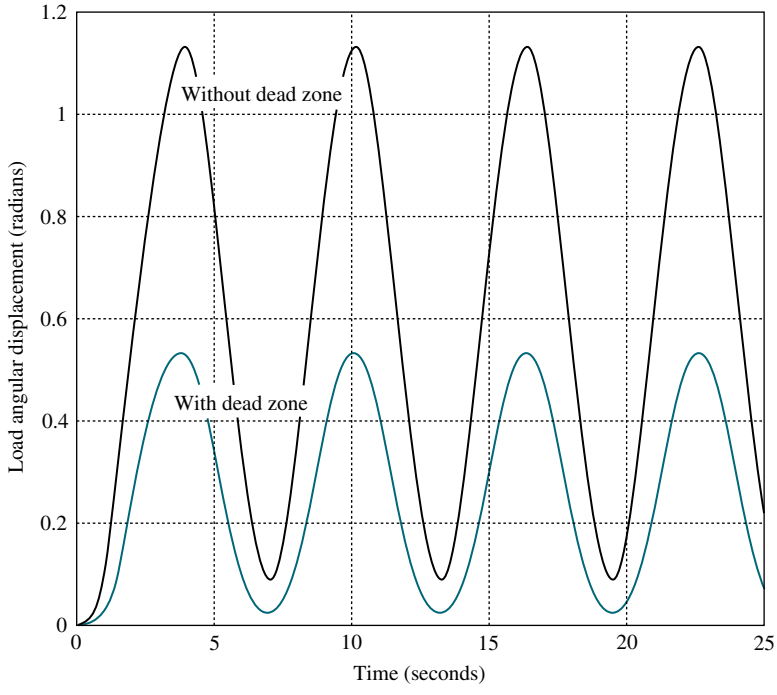
(*b*)

FIGURE 4.29 (*Continued*) **b.** Simulink block diagram

through an amplifier of unity gain that saturates at ±5 volts, Figure 4.29 shows that the effect of amplifier saturation is to limit the obtained velocity.

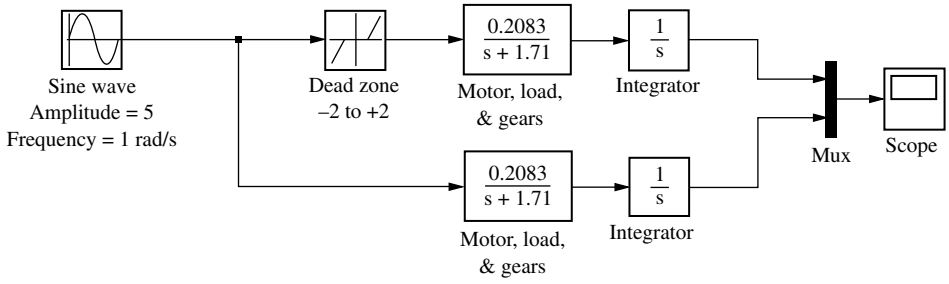
The effect of dead zone on the output shaft driven by a motor and gears is shown in Figure 4.30. Here we once again assume the motor, load, and gears from Antenna Control Case Study of Chapter 2. Dead zone is present when the motor cannot respond to small voltages. The motor input is a sinusoidal waveform chosen to allow us to see the effects of dead zone vividly. The response begins when the input voltage to the motor exceeds a threshold. We notice a lower amplitude when dead zone is present.

The effect of backlash on the output shaft driven by a motor and gears is shown in Figure 4.31. Again we assume the motor, load, and gears from the Antenna Control Case Study of Chapter 2. The motor input is again a sinusoidal waveform,



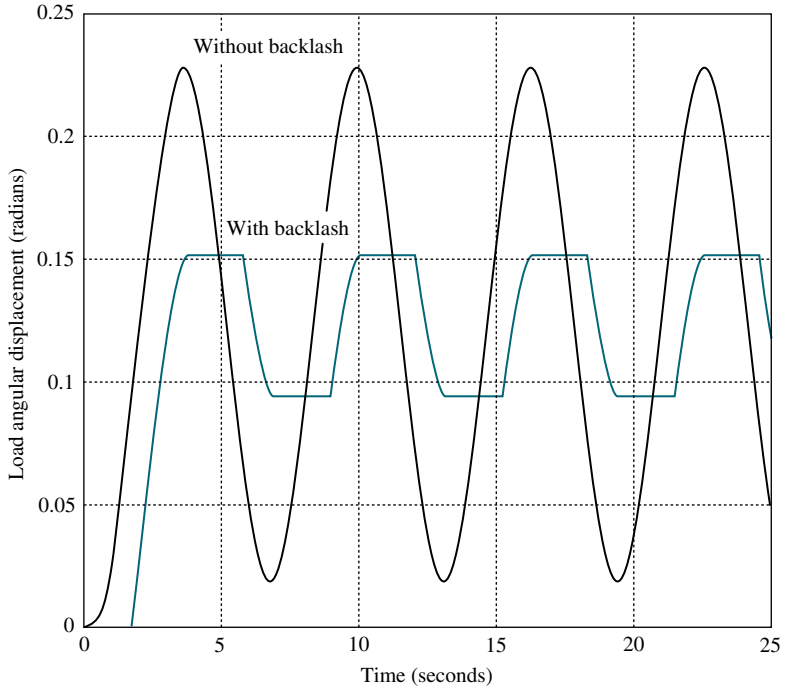
(*a*)

FIGURE 4.30 **a.** Effect of dead zone on load angular displacement response; (*figure continues*)

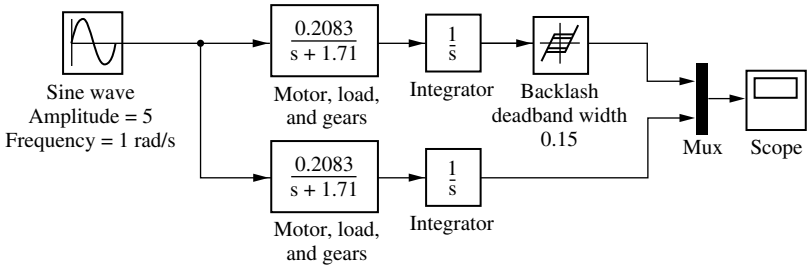


(*b*)

FIGURE 4.30 (*Continued*) **b.** Simulink block diagram



(*a*)



(*b*)

FIGURE 4.31 **a.** Effect of backlash on load angular displacement response; **b.** Simulink block diagram

which is chosen to allow us to see vividly the effects of backlash in the gears driven by the motor. As the motor reverses direction, the output shaft remains stationary while the motor begins to reverse. When the gears finally connect, the output shaft itself begins to turn in the reverse direction. The resulting response is quite different from the linear response without backlash.

Skill-Assessment Exercise 4.8

Simulink SL

PROBLEM: Use MATLAB's Simulink to reproduce Figure 4.31.

ANSWER: See Figure 4.31.

Now that we have seen the effects of nonlinearities on the time response, let us return to linear systems. Our coverage so far for linear systems has dealt with finding the time response by using the Laplace transform in the frequency domain. Another way to solve for the response is to use state-space techniques in the time domain. This topic is the subject of the next two sections.

4.10 Laplace Transform Solution of State Equations

State Space SS

In Chapter 3, systems were modeled in state space, where the state-space representation consisted of a state equation and an output equation. In this section, we use the Laplace transform to solve the state equations for the state and output vectors.

Consider the state equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.92}$$

and the output equation

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{4.93}$$

Taking the Laplace transform of both sides of the state equation yields

$$s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \tag{4.94}$$

In order to separate $\mathbf{X}(s)$, replace $s\mathbf{X}(s)$ with $s\mathbf{I}\mathbf{X}(s)$, where $\mathbf{I}$ is an $n \times n$ identity matrix, and n is the order of the system. Combining all of the $\mathbf{X}(s)$ terms, we get

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{x}(0) + \mathbf{B}\mathbf{U}(s) \tag{4.95}$$

Solving for $\mathbf{X}(s)$ by premultiplying both sides of Eq. (4.95) by $(s\mathbf{I} - \mathbf{A})^{-1}$, the final solution for $\mathbf{X}(s)$ is

$$\begin{aligned}\mathbf{X}(s) &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \\ &= \frac{\text{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})}[\mathbf{x}(0) + \mathbf{B}\mathbf{U}(s)]\end{aligned} \tag{4.96}$$

Taking the Laplace transform of the output equation yields

$$\boxed{\mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s)} \tag{4.97}$$

Eigenvalues and Transfer Function Poles

We saw that the poles of the transfer function determine the nature of the transient response of the system. Is there an equivalent quantity in the state-space representation that yields the same information? Section 5.8 formally defines the roots of $\det(s\mathbf{I} - \mathbf{A}) = 0$ (see the denominator of Eq. (4.96)) to be *eigenvalues* of the system matrix, $\mathbf{A}$.[6] Let us show that the eigenvalues are equal to the poles of the system's transfer function. Let the output, $\mathbf{Y}(s)$, and the input, $\mathbf{U}(s)$, be scalar quantities $Y(s)$ and $U(s)$, respectively. Further, to conform to the definition of a transfer function, let $\mathbf{x}(0)$, the initial state vector, equal $\mathbf{0}$, the null vector. Substituting Eq. (4.96) into Eq. (4.97) and solving for the transfer function, $Y(s)/U(s)$, yields

$$\begin{aligned}\frac{Y(s)}{U(s)} &= \mathbf{C}\left[\frac{\text{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})}\right]\mathbf{B} + \mathbf{D} \\ &= \frac{\mathbf{C}\,\text{adj}(s\mathbf{I} - \mathbf{A})\mathbf{B} + \mathbf{D}\det(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})}\end{aligned} \tag{4.98}$$

The roots of the denominator of Eq. (4.98) are the poles of the system. Since the denominators of Eqs. (4.96) and (4.98) are identical, the system poles equal the eigenvalues. Hence, if a system is represented in state-space, we can find the poles from $\det(s\mathbf{I} - \mathbf{A}) = 0$. We will be more formal about these facts when we discuss stability in Chapter 6.

The following example demonstrates solving the state equations using the Laplace transform as well as finding the eigenvalues and system poles.

Example 4.11

Laplace Transform Solution; Eigenvalues and Poles

PROBLEM: Given the system represented in state space by Eqs. (4.99),

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -24 & -26 & -9 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e^{-t} \tag{4.99a}$$

$$y = [1 \quad 1 \quad 0]\mathbf{x} \tag{4.99b}$$

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \tag{4.99c}$$

[6] Sometimes the symbol λ is used in place of the complex variable s when solving the state equations without using the Laplace transform. Thus, it is common to see the characteristic equation also written as $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$.

do the following:

a. Solve the preceding state equation and obtain the output for the given exponential input.

b. Find the eigenvalues and the system poles.

SOLUTION:

a. We will solve the problem by finding the component parts of Eq. (4.96), followed by substitution into Eq. (4.97). First obtain **A** and **B** by comparing Eq. (4.99a) to Eq. (4.92). Since

$$s\mathbf{I} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \tag{4.100}$$

then

$$(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 24 & 26 & s+9 \end{bmatrix} \tag{4.101}$$

and

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{\begin{bmatrix} (s^2+9s+26) & (s+9) & 1 \\ -24 & s^2+9s & s \\ -24s & -(26s+24) & s^2 \end{bmatrix}}{s^3+9s^2+26s+24} \tag{4.102}$$

Since $\mathbf{U}(s)$ (the Laplace transform for e^{-t}) is $1/(s+1)$, $\mathbf{X}(s)$ can be calculated. Rewriting Eq. (4.96) as

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}[\mathbf{x}(0) + \mathbf{B}\mathbf{U}(s)] \tag{4.103}$$

and using **B** and $\mathbf{x}(0)$ from Eqs. (4.99a) and (4.99c), respectively, we get

$$X_1(s) = \frac{(s^3+10s^2+37s+29)}{(s+1)(s+2)(s+3)(s+4)} \tag{4.104a}$$

$$X_2(s) = \frac{(2s^2-21s-24)}{(s+1)(s+2)(s+3)(s+4)} \tag{4.104b}$$

$$X_3(s) = \frac{s(2s^2-21s-24)}{(s+1)(s+2)(s+3)(s+4)} \tag{4.104c}$$

The output equation is found from Eq. (4.99b). Performing the indicated addition yields

$$Y(s) = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{bmatrix} = X_1(s) + X_2(s) \tag{4.105}$$

or

$$Y(s) = \frac{(s^3 + 12s^2 + 16s + 5)}{(s+1)(s+2)(s+3)(s+4)} = \frac{-6.5}{s+2} + \frac{19}{s+3} - \frac{11.5}{s+4} \tag{4.106}$$

where the pole at -1 canceled a zero at -1. Taking the inverse Laplace transform,

$$y(t) = -6.5e^{-2t} + 19e^{-3t} - 11.5e^{-4t} \tag{4.107}$$

b. The denominator of Eq. (4.102), which is $\det(s\mathbf{I} - \mathbf{A})$, is also the denominator of the system's transfer function. Thus, $\det(s\mathbf{I} - \mathbf{A}) = 0$ furnishes both the poles of the system and the eigenvalues -2, -3, and -4.

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch4sp1 in Appendix F at www.wiley.com/college/nise. You will learn how to solve state equations for the output response using the Laplace transform. Example 4.11 will be solved using MATLAB and the Symbolic Math Toolbox.

Skill-Assessment Exercise 4.9

WileyPLUS WPCS Control Solutions

PROBLEM: Given the system represented in state space by Eqs. (4.108),

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 2 \\ -3 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{-t} \tag{4.108a}$$

$$y = [1 \quad 3]\mathbf{x} \tag{4.108b}$$

$$\mathbf{x}(0) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{4.108c}$$

do the following:

a. Solve for $y(t)$ using state-space and Laplace transform techniques.

b. Find the eigenvalues and the system poles.

ANSWERS:

a. $y(t) = -0.5e^{-t} - 12e^{-2t} + 17.5e^{-3t}$

b. -2, -3

The complete solution is located at www.wiley.com/college/nise.

TryIt 4.5

Use the following MATLAB and Symbolic Math Toolbox statements to solve Skill-Assessment Exercise 4.9.

```
Syms s
A=[0 2;-3 -5]; B=[0;1];
C=[1 3];X0=[2;1];
U=1/(s+1);
I=[1 0;0 1];
X=((s*I-A)^-1)*...
 (X0+B*U);
Y=C*X;Y=simplify(Y);
y=ilaplace(Y);
pretty(y)
eig(A)
```

4.11 Time Domain Solution of State Equations

State Space
SS

We now look at another technique for solving the state equations. Rather than using the Laplace transform, we solve the equations directly in the time domain using a method closely allied to the classical solution of differential equations. We will find that the final solution consists of two parts that are different from the forced and natural responses.

The solution in the time domain is given directly by

$$\begin{aligned}\mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \\ &= \mathbf{\Phi}(t)\mathbf{x}(0) + \int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau\end{aligned} \tag{4.109}$$

where $\mathbf{\Phi}(t) = e^{\mathbf{A}t}$ by definition, and which is called the *state-transition matrix*. Eq. (4.109) is derived in Appendix I located at www.wiley.com/college/nise. Readers who are not familiar with this equation or who may want to refresh their memory should consult Appendix I before proceeding.

Notice that the first term on the right-hand side of the equation is the response due to the initial state vector, $\mathbf{x}(0)$. Notice also that it is the only term dependent on the initial state vector and not the input. We call this part of the response the *zero-input response*, since it is the total response if the input is zero. The second term, called the *convolution integral*, is dependent only on the input, $\mathbf{u}$, and the input matrix, $\mathbf{B}$, not the initial state vector. We call this part of the response the *zero-state response*, since it is the total response if the initial state vector is zero. Thus, there is a partitioning of the response different from the forced/natural response we have seen when solving differential equations. In differential equations, the arbitrary constants of the natural response are evaluated based on the initial conditions and the initial values of the forced response and its derivatives. Thus, the natural response's amplitudes are a function of the initial conditions of the output and the input. In Eq. (4.109), the zero-input response is not dependent on the initial values of the input and its derivatives. It is dependent only on the initial conditions of the state vector. The next example vividly shows the difference in partitioning. Pay close attention to the fact that in the final result the zero-state response contains not only the forced solution but also pieces of what we previously called the natural response. We will see in the solution that the natural response is distributed through the zero-input response and the zero-state response.

Before proceeding with the example, let us examine the form the elements of $\mathbf{\Phi}(t)$ take for linear, time-invariant systems. The first term of Eq. (4.96), the Laplace transform of the response for unforced systems, is the transform of $\Phi(t)\mathbf{x}(0)$, the zero-input response from Eq. (4.109). Thus, for the unforced system

$$\mathscr{L}[\mathbf{x}(t)] = \mathscr{L}[\mathbf{\Phi}(t)\mathbf{x}(0)] = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) \tag{4.110}$$

from which we can see that $(s\mathbf{I} - \mathbf{A})^{-1}$ *is the Laplace transform of the state-transition matrix*, $\mathbf{\Phi}(t)$. We have already seen that the denominator of $(s\mathbf{I} - \mathbf{A})^{-1}$ is a polynomial in s whose roots are the system poles. This polynomial is found from

the equation $\det(s\mathbf{I} - \mathbf{A}) = 0$. Since

$$\mathscr{L}^{-1}[(s\mathbf{I} - \mathbf{A})^{-1}] = \mathscr{L}^{-1}\left[\frac{\text{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})}\right] = \mathbf{\Phi}(t) \tag{4.111}$$

each term of $\Phi(t)$ would be the sum of exponentials generated by the system's poles.

Let us summarize the concepts with two numerical examples. The first example solves the state equations directly in the time domain. The second example uses the Laplace transform to solve for the state-transition matrix by finding the inverse Laplace transform of $(s\mathbf{I} - \mathbf{A})^{-1}$.

Example 4.12

Time Domain Solution

PROBLEM: For the state equation and initial state vector shown in Eqs. (4.112), where $u(t)$ is a unit step, find the state-transition matrix and then solve for $\mathbf{x}(t)$.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -8 & -6 \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(t) \tag{4.112a}$$

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4.112b}$$

SOLUTION: Since the state equation is in the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \tag{4.113}$$

find the eigenvalues using $\det(s\mathbf{I} - \mathbf{A}) = 0$. Hence, $s^2 + 6s + 8 = 0$, from which $s_1 = -2$ and $s_2 = -4$. Since each term of the state-transition matrix is the sum of responses generated by the poles (eigenvalues), we assume a state-transition matrix of the form

$$\mathbf{\Phi}(t) = \begin{bmatrix} (K_1e^{-2t} + K_2e^{-4t}) & (K_3e^{-2t} + K_4e^{-4t}) \\ (K_5e^{-2t} + K_6e^{-4t}) & (K_7e^{-2t} + K_8e^{-4t}) \end{bmatrix} \tag{4.114}$$

In order to find the values of the constants, we make use of the properties of the state-transition matrix derived in Appendix J located at www.wiley.com/college/nise.

$$\mathbf{\Phi}(0) = \mathbf{I} \tag{4.115}$$

$$K_1 + K_2 = 1 \tag{4.116a}$$

$$K_3 + K_4 = 0 \tag{4.116b}$$

$$K_5 + K_6 = 0 \tag{4.116c}$$

$$K_7 + K_8 = 1 \tag{4.116d}$$

and since

$$\dot{\mathbf{\Phi}}(0) = \mathbf{A} \tag{4.117}$$

then

$$-2K_1 - 4K_2 = 0 \tag{4.118a}$$

$$-2K_3 - 4K_4 = 1 \tag{4.118b}$$

$$-2K_5 - 4K_6 = -8 \tag{4.118c}$$

$$-2K_7 - 4K_8 = -6 \tag{4.118d}$$

The constants are solved by taking two simultaneous equations four times. For example, Eq. (4.116*a*) can be solved simultaneously with Eq. (4.118*a*) to yield the values of K_1 and K_2. Proceeding similarly, all of the constants can be found. Therefore,

$$\mathbf{\Phi}(t) = \begin{bmatrix} (2e^{-2t} - e^{-4t}) & \left(\frac{1}{2}e^{-2t} - \frac{1}{2}e^{-4t}\right) \\ (-4e^{-2t} + 4e^{-4t}) & (-e^{-2t} + 2e^{-4t}) \end{bmatrix} \tag{4.119}$$

Also,

$$\mathbf{\Phi}(t-\tau)\mathbf{B} = \begin{bmatrix} \left(\frac{1}{2}e^{-2(t-\tau)} - \frac{1}{2}e^{-4(t-\tau)}\right) \\ \left(-e^{-2(t-\tau)} + 2e^{-4(t-\tau)}\right) \end{bmatrix} \tag{4.120}$$

Hence, the first term of Eq. (4.109) is

$$\mathbf{\Phi}(t)\mathbf{x}(0) = \begin{bmatrix} (2e^{-2t} - e^{-4t}) \\ (-4e^{-2t} + 4e^{-4t}) \end{bmatrix} \tag{4.121}$$

The last term of Eq. (4.109) is

$$\begin{aligned} \int_0^t \mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau &= \begin{bmatrix} \frac{1}{2}e^{-2t}\int_0^t e^{2\tau}d\tau - \frac{1}{2}e^{-4t}\int_0^t e^{4\tau}d\tau \\ -e^{-2t}\int_0^t e^{2\tau}d\tau + 2e^{-4t}\int_0^t e^{4\tau}d\tau \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{8} - \frac{1}{4}e^{-2t} + \frac{1}{8}e^{-4t} \\ \frac{1}{2}e^{-2t} - \frac{1}{2}e^{-4t} \end{bmatrix} \end{aligned} \tag{4.122}$$

Notice, as promised, that Eq. (4.122), the zero-state response, contains not only the forced response, 1/8, but also terms of the form Ae^{-2t} and Be^{-4t} that are part of

what we previously called the natural response. However, the coefficients, A and B, are not dependent on the initial conditions.

The final result is found by adding Eqs. (4.121) and (4.122). Hence,

$$\mathbf{x}(t) = \mathbf{\Phi}(t)\mathbf{x}(0) + \int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau = \begin{bmatrix} \frac{1}{8} + \frac{7}{4}e^{-2t} - \frac{7}{8}e^{-4t} \\ -\frac{7}{2}e^{-2t} + \frac{7}{2}e^{-4t} \end{bmatrix} \tag{4.123}$$

Example 4.13

State-Transition Matrix via Laplace Transform

PROBLEM: Find the state-transition matrix of Example 4.12, using $(s\mathbf{I} - \mathbf{A})^{-1}$.

SOLUTION: We use the fact that $\mathbf{\Phi}(t)$ is the inverse Laplace transform of $(s\mathbf{I} - \mathbf{A})^{-1}$. Thus, first find $(s\mathbf{I} - \mathbf{A})$ as

$$(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s & -1 \\ 8 & (s+6) \end{bmatrix} \tag{4.124}$$

from which

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{\begin{bmatrix} s+6 & 1 \\ -8 & s \end{bmatrix}}{s^2 + 6s + 8} = \begin{bmatrix} \frac{s+6}{s^2+6s+8} & \frac{1}{s^2+6s+8} \\ \frac{-8}{s^2+6s+8} & \frac{s}{s^2+6s+8} \end{bmatrix} \tag{4.125}$$

Expanding each term in the matrix on the right by partial fractions yields

$$(s\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} \left(\frac{2}{s+2} - \frac{1}{s+4}\right) & \left(\frac{1/2}{s+2} - \frac{1/2}{s+4}\right) \\ \left(\frac{-4}{s+2} + \frac{4}{s+4}\right) & \left(\frac{-1}{s+2} + \frac{2}{s+4}\right) \end{bmatrix} \tag{4.126}$$

Finally, taking the inverse Laplace transform of each term, we obtain

$$\mathbf{\Phi}(t) = \begin{bmatrix} (2e^{-2t} - e^{-4t}) & \left(\frac{1}{2}e^{-2t} - \frac{1}{2}e^{-4t}\right) \\ (-4e^{-2t} + 4e^{-4t}) & (-e^{-2t} + 2e^{-4t}) \end{bmatrix} \tag{4.127}$$

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch4sp2 in Appendix F at www.wiley.com/college/nise. You will learn how to solve state equations for the output response using the convolution integral. Examples 4.12 and 4.13 will be solved using MATLAB and the Symbolic Math Toolbox.

Systems represented in state space can be simulated on the digital computer. Programs such as MATLAB can be used for this purpose. Alternately, the user can write specialized programs, as discussed in Appendix G.1 at www.wiley.com/college/nise.

MATLAB
ML

`Students who are using MATLAB should now run ch4p3 in Appendix B. This exercise uses MATLAB to simulate the step response of systems represented in state space. In addition to generating the step response, you will learn how to specify the range on the time axis for the plot.`

Skill-Assessment Exercise 4.10

WileyPLUS
WPCS
Control Solutions

PROBLEM: Given the system represented in state space by Eqs. (4.128):

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 2 \\ -2 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e^{-2t} \tag{4.128a}$$

$$y = \begin{bmatrix} 2 & 1 \end{bmatrix}\mathbf{x} \tag{4.128b}$$

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{4.128c}$$

do the following:

a. Solve for the state-transition matrix.
b. Solve for the state vector using the convolution integral.
c. Find the output, $y(t)$.

ANSWERS:

a. $\mathbf{\Phi}(t) = \begin{bmatrix} \left(\frac{4}{3}e^{-t} - \frac{1}{3}e^{-4t}\right) & \left(\frac{2}{3}e^{-t} - \frac{2}{3}e^{-4t}\right) \\ \left(-\frac{2}{3}e^{-t} + \frac{2}{3}e^{-4t}\right) & \left(-\frac{1}{3}e^{-t} + \frac{4}{3}e^{-4t}\right) \end{bmatrix}$

b. $\mathbf{x}(t) = \begin{bmatrix} \left(\frac{10}{3}e^{-t} - e^{-2t} - \frac{4}{3}e^{-4t}\right) \\ \left(-\frac{5}{3}e^{-t} + e^{-2t} + \frac{8}{3}e^{-4t}\right) \end{bmatrix}$

c. $y(t) = 5e^{-t} - e^{-2t}$

The complete solution is located at www.wiley.com/college/nise.

Case Studies

Antenna Control: Open-Loop Response

In this chapter, we have made use of the transfer functions derived in Chapter 2 and the state equations derived in Chapter 3 to obtain the output response of an

open-loop system. We also showed the importance of the poles of a system in determining the transient response. The following case study uses these concepts to analyze an open-loop portion of the antenna azimuth position control system. The open-loop function that we will deal with consists of a power amplifier and motor with load.

PROBLEM: For the schematic of the azimuth position control system shown on the front endpapers, Configuration 1, assume an open-loop system (feedback path disconnected).

a. Predict, by inspection, the form of the open-loop angular velocity response of the load to a step-voltage input to the power amplifier.

b. Find the damping ratio and natural frequency of the open-loop system.

c. Derive the complete analytical expression for the open-loop angular velocity response of the load to a step-voltage input to the power amplifier, using transfer functions.

State Space SS

d. Obtain the open-loop state and output equations.

MATLAB ML

e. `Use MATLAB to obtain a plot of the open-loop angular velocity response to a step-voltage input.`

SOLUTION: The transfer functions of the power amplifier, motor, and load as shown on the front endpapers, Configuration 1, were discussed in the Chapter 2 case study. The two subsystems are shown interconnected in Figure 4.32(*a*). Differentiating the angular position of the motor and load output by multiplying by s, we obtain the output angular velocity, ω_o, as shown in Figure 4.32(*a*). The equivalent transfer function representing the three blocks in Figure 4.32(*a*) is the product of the individual transfer functions and is shown in Figure 4.32(b).[7]

a. Using the transfer function shown in Figure 4.32(*b*), we can predict the nature of the step response. The step response consists of the steady-state response generated by the step input and the transient response, which is the sum of two exponentials generated by each pole of the transfer function. Hence, the form of the response is

$$\omega_o(t) = A + Be^{-100t} + Ce^{-1.71t} \tag{4.129}$$

b. The damping ratio and natural frequency of the open-loop system can be found by expanding the denominator of the transfer function. Since the open-loop

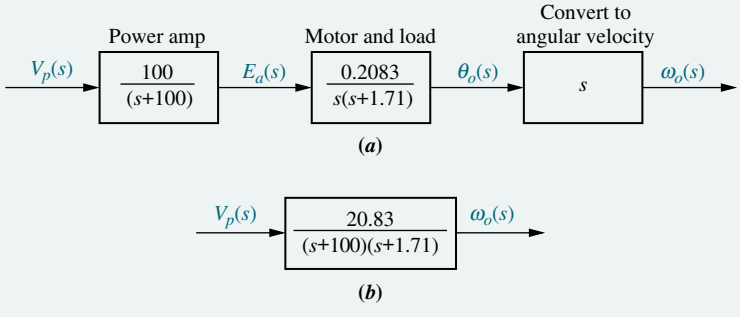


FIGURE 4.32 Antenna azimuth position control system for angular velocity: **a.** forward path; **b.** equivalent forward path

[7] This product relationship will be derived in Chapter 5.

transfer function is

$$G(s) = \frac{20.83}{s^2 + 101.71s + 171} \tag{4.130}$$

$\omega_n = \sqrt{171} = 13.08$, and $\zeta = 3.89$ (overdamped).

c. In order to derive the angular velocity response to a step input, we multiply the transfer function of Eq. (4.130) by a step input, $1/s$, and obtain

$$\omega_o(s) = \frac{20.83}{s(s + 100)(s + 1.71)} \tag{4.131}$$

Expanding into partial fractions, we get

$$\omega_o(s) = \frac{0.122}{s} + \frac{2.12 \times 10^{-3}}{s + 100} - \frac{0.124}{s + 1.71} \tag{4.132}$$

Transforming to the time domain yields

$$\omega_o(t) = 0.122 + (2.12 \times 10^{-3})e^{-100t} - 0.124e^{-1.71t} \tag{4.133}$$

d. First convert the transfer function into the state-space representation. Using Eq. (4.130), we have

State Space SS

$$\frac{\omega_o(s)}{V_p(s)} = \frac{20.83}{s^2 + 101.71s + 171} \tag{4.134}$$

Cross-multiplying and taking the inverse Laplace transform with zero initial conditions, we have

$$\ddot{\omega}_o + 101.71\dot{\omega}_o + 171\omega_o = 20.83v_p \tag{4.135}$$

Defining the phase variables as

$$x_1 = \omega_o \tag{4.136a}$$

$$x_2 = \dot{\omega}_o \tag{4.136b}$$

and using Eq. (4.135), the state equations are written as

$$\dot{x}_1 = x_2 \tag{4.137a}$$

$$\dot{x}_2 = -171x_1 - 101.71x_2 + 20.83v_p \tag{4.137b}$$

where $v_p = 1$, a unit step. Since $x_1 = \omega_o$ is the output, the output equation is

$$y = x_1 \tag{4.138}$$

Equations (4.137) and (4.138) can be programmed to obtain the step response using MATLAB or alternative methods described in Appendix H.1 at www.wiley.com/college/nise.

e. `Students who are using MATLAB should now run ch4p4 in Appendix B. This exercise uses MATLAB to plot the step response.`

MATLAB ML

CHALLENGE: You are now given a problem to test your knowledge of this chapter's objectives. Refer to the antenna azimuth position control system shown on the

front endpapers, Configuration 2. Assume an open-loop system (feedback path disconnected) and do the following:

a. Predict the open-loop angular velocity response of the power amplifier, motor, and load to a step voltage at the input to the power amplifier.

b. Find the damping ratio and natural frequency of the open-loop system.

c. Derive the open-loop angular velocity response of the power amplifier, motor, and load to a step-voltage input using transfer functions.

State Space SS

d. Obtain the open-loop state and output equations.

MATLAB ML

e. `Use MATLAB to obtain a plot of the open-loop angular velocity response to a step-voltage input.`

Unmanned Free-Swimming Submersible Vehicle: Open-Loop Pitch Response

An Unmanned Free-Swimming Submersible (UFSS) vehicle is shown in Figure 4.33. The depth of the vehicle is controlled as follows. During forward motion, an elevator surface on the vehicle is deflected by a selected amount. This deflection causes the vehicle to rotate about the pitch axis. The pitch of the vehicle creates a vertical force that causes the vehicle to submerge or rise. The pitch control system for the vehicle is used here and in subsequent chapters as a case study to demonstrate the covered concepts. The block diagram for the pitch control system is shown in Figure 4.34 and on the back endpapers for future reference (*Johnson, 1980*). In this case study, we investigate the time response of the vehicle dynamics that relate the pitch angle output to the elevator deflection input.

PROBLEM: The transfer function relating pitch angle, $\theta(s)$, to elevator surface angle, $\delta_e(s)$, for the UFSS vehicle is

$$\frac{\theta(s)}{\delta_e(s)} = \frac{-0.125(s+0.435)}{(s+1.23)(s^2+0.226s+0.0169)} \tag{4.139}$$

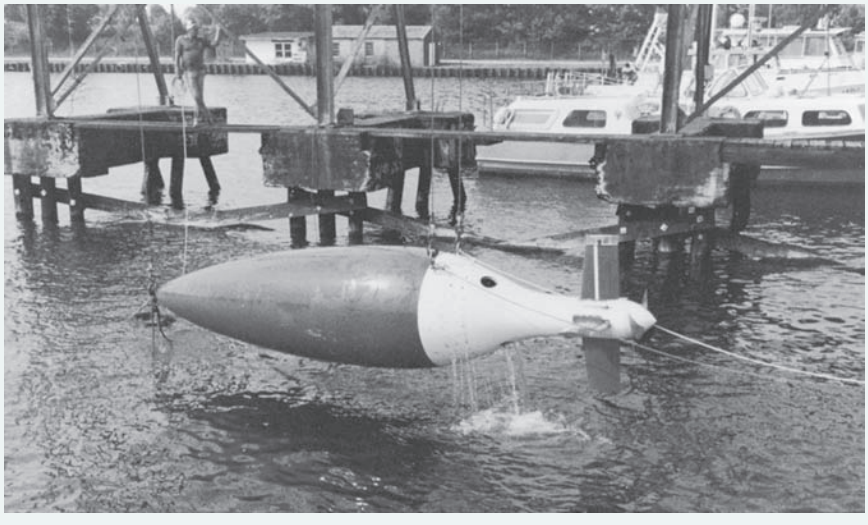


FIGURE 4.33 Unmanned Free-Swimming Submersible (UFSS) vehicle.

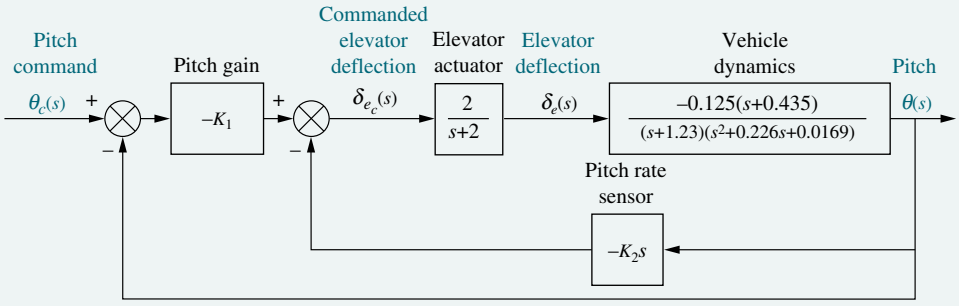


FIGURE 4.34 Pitch control loop for the UFSS vehicle

a. Using only the second-order poles shown in the transfer function, predict percent overshoot, rise time, peak time, and settling time.

b. Using Laplace transforms, find the analytical expression for the response of the pitch angle to a step input in elevator surface deflection.

c. Evaluate the effect of the additional pole and zero on the validity of the second-order approximation.

d. Plot the step response of the vehicle dynamics and verify your conclusions found in (c).

SOLUTION:

a. Using the polynomial $s^2 + 0.226s + 0.0169$, we find that $\omega_n^2 = 0.0169$ and $2\zeta\omega_n = 0.226$. Thus, $\omega_n = 0.13$ rad/s and $\zeta = 0.869$. Hence, $\%OS = e^{-\zeta\pi/\sqrt{1-\zeta^2}}100 = 0.399\%$. From Figure 4.16, $\omega_n T_r = 2.75$, or $T_r = 21.2$ s. To find peak time, we use $T_p = \pi/\omega_n\sqrt{1-\zeta^2} = 48.9$ s. Finally, settling time is $T_s = 4/\zeta\omega_n = 35.4$ s.

b. In order to display a positive final value in Part **d**. we find the response of the system to a negative unit step, compensating for the negative sign in the transfer function. Using partial-fraction expansion, the Laplace transform of the response, $\theta(s)$, is

$$\theta(s) = \frac{0.125(s+0.435)}{s(s+1.23)(s^2+0.226s+0.0169)}$$

$$= 2.616\frac{1}{s} + 0.0645\frac{1}{s+1.23}$$

$$- \frac{2.68(s+0.113) + 3.478\sqrt{0.00413}}{(s+0.113)^2 + 0.00413} \qquad (4.140)$$

Taking the inverse Laplace transform,

$$\theta(t) = 2.616 + 0.0645e^{-1.23t}$$

$$- e^{-0.113t}(2.68\cos 0.0643t + 3.478\sin 0.0643t)$$

$$= 2.616 + 0.0645e^{-1.23t} - 4.39e^{-0.113t}\cos(0.0643t + 52.38^\circ) \qquad (4.141)$$

c. Looking at the relative amplitudes between the coefficient of the $e^{-1.23t}$ term and the cosine term in Eq. (4.165), we see that there is pole-zero cancellation between the pole at -1.23 and the zero at -0.435. Further, the pole at -1.23 is more than five times farther from the $j\omega$ axis than the second-order dominant

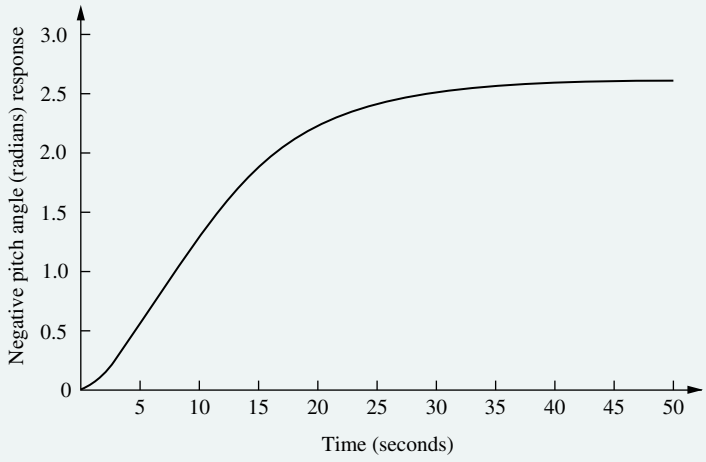


FIGURE 4.35 Negative step response of pitch control for UFSS vehicle

poles at $-0.113 \pm j0.0643$. We conclude that the response will be close to that predicted.

d. Plotting Eq. (4.141) or using a computer simulation, we obtain the step response shown in Figure 4.35. We indeed see a response close to that predicted.

MATLAB ML

Students who are using MATLAB should now run ch4p5 in Appendix B. This exercise uses MATLAB to find ζ, ω_n, T_s, T_p, and T_r and plot a step response. Table lookup is used to find T_r. The exercise applies the concepts to the problem above.

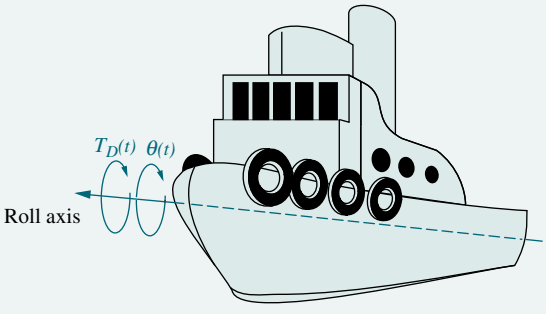


FIGURE 4.36 A ship at sea, showing roll axis

CHALLENGE: You are now given a problem to test your knowledge of this chapter's objectives. This problem uses the same principles that were applied to the Unmanned Free-Swimming Submersible vehicle: Ships at sea undergo motion about their roll axis, as shown in Figure 4.36. Fins called *stabilizers* are used to reduce this rolling motion. The stabilizers can be positioned by a closed-loop roll control system that consists of components, such as fin actuators and sensors, as well as the ship's roll dynamics.

Assume the roll dynamics, which relates the roll-angle output, $\theta(s)$, to a disturbance-torque input, $T_D(s)$, is

$$\frac{\theta(s)}{T_D(s)} = \frac{2.25}{(s^2 + 0.5s + 2.25)} \tag{4.142}$$

Do the following:

a. Find the natural frequency, damping ratio, peak time, settling time, rise time, and percent overshoot.

b. Find the analytical expression for the output response to a unit step input in voltage.

MATLAB ML

c. Use MATLAB to solve **a** and **b** and to plot the response found in **b**.

Summary

In this chapter, we took the system models developed in Chapters 2 and 3 and found the output response for a given input, usually a step. The step response yields a clear picture of the system's transient response. We performed this analysis for two types of systems, *first order* and *second order*, which are representative of many physical systems. We then formalized our findings and arrived at numerical specifications describing the responses.

For first-order systems having a single pole on the real axis, the specification of transient response that we derived was the *time constant*, which is the reciprocal of the real-axis pole location. This specification gives us an indication of the speed of the transient response. In particular, the time constant is the time for the step response to reach 63% of its final value.

Second-order systems are more complex. Depending on the values of system components, a second-order system can exhibit four kinds of behavior:

1. Overdamped
2. Underdamped
3. Undamped
4. Critically damped

We found that the poles of the input generate the forced response, whereas the system poles generate the transient response. If the system poles are real, the system exhibits *overdamped* behavior. These exponential responses have time constants equal to the reciprocals of the pole locations. Purely imaginary poles yield *undamped* sinusoidal oscillations whose radian frequency is equal to the magnitude of the imaginary pole. Systems with complex poles display *underdamped* responses. The real part of the complex pole dictates the exponential decay envelope, and the imaginary part dictates the sinusoidal radian frequency. The exponential decay envelope has a time constant equal to the reciprocal of the real part of the pole, and the sinusoid has a radian frequency equal to the imaginary part of the pole.

For all second-order cases, we developed specifications called the *damping ratio*, ζ, and *natural frequency*, ω_n. The damping ratio gives us an idea about the nature of the transient response and how much overshoot and oscillation it undergoes, regardless of time scaling. The natural frequency gives an indication of the speed of the response.

We found that the value of ζ determines the form of the second-order natural response:

- If $\zeta = 0$, the response is undamped.
- If $\zeta < 1$, the response is underdamped.
- If $\zeta = 1$, the response is critically damped.
- If $\zeta > 1$, the response is overdamped.

The natural frequency is the frequency of oscillation if all damping is removed. It acts as a scaling factor for the response, as can be seen from Eq. (4.28), in which the independent variable can be considered to be $\omega_n t$.

For the underdamped case we defined several transient response specifications, including these:

- Percent overshoot, $\%OS$
- Peak time, T_p
- Settling time, T_s
- Rise time, T_r

The peak time is inversely proportional to the imaginary part of the complex pole. Thus, horizontal lines on the *s*-plane are lines of constant peak time. Percent overshoot is a function of only the damping ratio. Consequently, radial lines are lines of constant percent overshoot. Finally, settling time is inversely proportional to the real part of the complex pole. Hence, vertical lines on the *s*-plane are lines of constant settling time.

We found that peak time, percent overshoot, and settling time are related to pole location. Thus, we can design transient responses by relating a desired response to a pole location and then relating that pole location to a transfer function and the system's components.

The effects of nonlinearities, such as saturation, dead zone, and backlash were explored using MATLAB's Simulink.

In this chapter, we also evaluated the time response using the state-space approach. The response found in this way was separated into the *zero-input response*, and the *zero-state response*, whereas the frequency response method yielded a total response divided into *natural response* and *forced response* components.

In the next chapter we will use the transient response specifications developed here to analyze and design systems that consist of the interconnection of multiple subsystems. We will see how to reduce these systems to a single transfer function in order to apply the concepts developed in Chapter 4.

Review Questions

1. Name the performance specification for first-order systems.
2. What does the performance specification for a first-order system tell us?
3. In a system with an input and an output, what poles generate the steady-state response?
4. In a system with an input and an output, what poles generate the transient response?
5. The imaginary part of a pole generates what part of a response?
6. The real part of a pole generates what part of a response?
7. What is the difference between the natural frequency and the damped frequency of oscillation?
8. If a pole is moved with a constant imaginary part, what will the responses have in common?
9. If a pole is moved with a constant real part, what will the responses have in common?
10. If a pole is moved along a radial line extending from the origin, what will the responses have in common?

11. List five specifications for a second-order underdamped system.

12. For Question 11 how many specifications completely determine the response?

13. What pole locations characterize (1) the underdamped system, (2) the overdamped system, and (3) the critically damped system?

14. Name two conditions under which the response generated by a pole can be neglected.

15. How can you justify pole-zero cancellation?

16. Does the solution of the state equation yield the output response of the system? Explain. State Space SS

17. What is the relationship between $(s\mathbf{I} - \mathbf{A})$, which appeared during the Laplace transformation solution of the state equations, and the state-transition matrix, which appeared during the classical solution of the state equation? State Space SS

18. Name a major advantage of using time-domain techniques for the solution of the response. State Space SS

19. Name a major advantage of using frequency-domain techniques for the solution of the response.

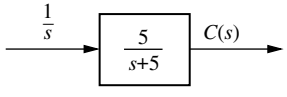
20. What three pieces of information must be given in order to solve for the output response of a system using state-space techniques? State Space SS

21. How can the poles of a system be found from the state equations? State Space SS

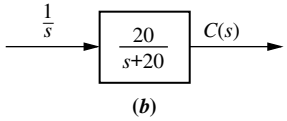
Problems

1. Derive the output responses for all parts of Figure 4.7. [Section: 4.4]

2. Find the output response, $c(t)$, for each of the systems shown in Figure P4.1. Also find the time constant, rise time, and settling time for each case. [Sections: 4.2, 4.3] WileyPLUS WPCS Control Solutions



(a)



(b)

FIGURE P4.1

3. Plot the step responses for Problem 2 using MATLAB. MATLAB ML

4. Find the capacitor voltage in the network shown in Figure P4.2 if the switch closes at $t = 0$. Assume zero initial conditions. Also find the time constant, rise time, and settling time for the capacitor voltage. [Sections: 4.2, 4.3]

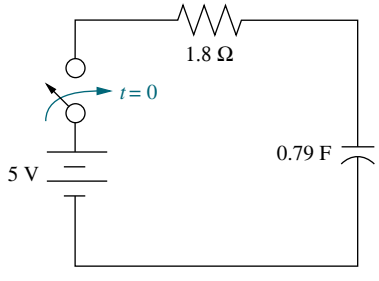


FIGURE P4.2

5. Plot the step response for Problem 4 using MATLAB. From your plots, find the time constant, rise time, and settling time. MATLAB ML

6. For the system shown in Figure P4.3, (a) find an equation that relates settling time of the velocity of

the mass to M; (b) find an equation that relates rise time of the velocity of the mass to M. [Sections: 4.2, 4.3]

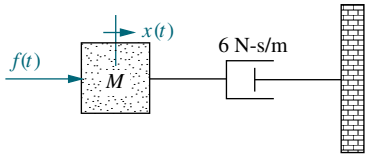


FIGURE P4.3

7. Plot the step response for Problem 6 using MATLAB. From your plots, find the time constant, rise time, and settling time. Use $M = 1$ and $M = 2$.

MATLAB ML

8. For each of the transfer functions shown below, find the locations of the poles and zeros, plot them on the s-plane, and then write an expression for the general form of the step response without solving for the inverse Laplace transform. State the nature of each response (overdamped, underdamped, and so on). [Sections: 4.3, 4.4]

a. $T(s) = \dfrac{2}{s+2}$

b. $T(s) = \dfrac{5}{(s+3)(s+6)}$

c. $T(s) = \dfrac{10(s+7)}{(s+10)(s+20)}$

d. $T(s) = \dfrac{20}{s^2+6s+144}$

e. $T(s) = \dfrac{s+2}{s^2+9}$

f. $T(s) = \dfrac{(s+5)}{(s+10)^2}$

9. Use MATLAB to find the poles of [Section: 4.2]

$$T(s) = \frac{s^2+2s+2}{s^4+6s^3+4s^2+7s+2}$$

MATLAB ML

10. Find the transfer function and poles of the system represented in state space here: [Section: 4.10]

$$\dot{\mathbf{x}} = \begin{bmatrix} 8 & -4 & 1 \\ -3 & 2 & 0 \\ 5 & 7 & -9 \end{bmatrix}\mathbf{x} + \begin{bmatrix} -4 \\ -3 \\ 4 \end{bmatrix} u(t)$$

$$y = \begin{bmatrix} 2 & 8 & -3 \end{bmatrix}\mathbf{x};\ \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

11. Repeat Problem 10 using MATLAB. [Section: 4.10].

MATLAB ML

12. Write the general form of the capacitor voltage for the electrical network shown in Figure P4.4. [Section: 4.4].

WileyPLUS WPCS Control Solutions

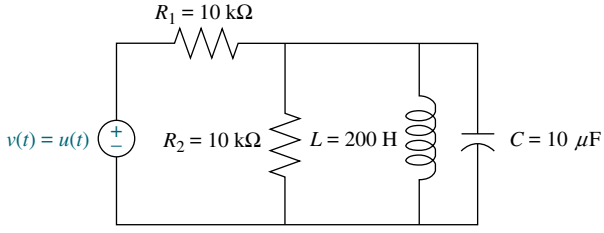


FIGURE P4.4

13. Use MATLAB to plot the capacitor voltage in Problem 12. [Section: 4.4].

MATLAB ML

14. Solve for $x(t)$ in the system shown in Figure P4.5 if $f(t)$ is a unit step. [Section: 4.4].

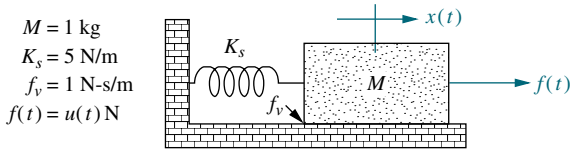


FIGURE P4.5

15. The system shown in Figure P4.6 has a unit step input. Find the output response as a function of time. Assume the system is underdamped. Notice that the result will be Eq. (4.28). [Section: 4.6].

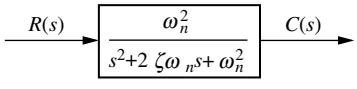


FIGURE P4.6

16. Derive the relationship for damping ratio as a function of percent overshoot, Eq. (4.39). [Section: 4.6].

17. Calculate the exact response of each system of Problem 8 using Laplace transform techniques, and compare the results to those obtained in that problem. [Sections: 4.3, 4.4].

18. Find the damping ratio and natural frequency for each second-order system of Problem 8 and show that the value of the damping ratio conforms to the type of response (underdamped, overdamped, and so on) predicted in that problem. [Section: 4.5].

19. A system has a damping ratio of 0.5, a natural frequency of 100 rad/s, and a dc gain of 1. Find the response of the system to a unit step input. [Section: 4.6].

WileyPLUS WPCS Control Solutions

20. For each of the second-order systems that follow, find ζ, ω_n, T_s, T_p, T_r, and $\%OS$. [Section: 4.6].

a. $T(s) = \dfrac{16}{s^2 + 3s + 16}$

b. $T(s) = \dfrac{0.04}{s^2 + 0.02s + 0.04}$

c. $T(s) = \dfrac{1.05 \times 10^7}{s^2 + 1.6 \times 10^3 s + 1.05 \times 10^7}$

21. Repeat Problem 20 using MATLAB. Have the computer program estimate the given specifications and plot the step responses. Estimate the rise time from the plots. [Section: 4.6].

MATLAB ML

22. Use MATLAB's LTI Viewer and obtain settling time, peak time, rise time, and percent overshoot for each of the systems in Problem 20. [Section: 4.6].

Gui Tool GUIT

23. For each pair of second-order system specifications that follow, find the location of the second-order pair of poles. [Section: 4.6].

a. $\%OS = 12\%$; $T_s = 0.6$ second

b. $\%OS = 10\%$; $T_p = 5$ seconds

c. $T_s = 7$ seconds; $T_p = 3$ seconds

24. Find the transfer function of a second-order system that yields a 12.3% overshoot and a settling time of 1 second. [Section: 4.6]

WileyPLUS WPCS Control Solutions

25. For the system shown in Figure P4.7, do the following: [Section: 4.6]

a. Find the transfer function $G(s) = X(s)/F(s)$.

b. Find ζ, ω_n, $\%OS$, T_s, T_p, and T_r.

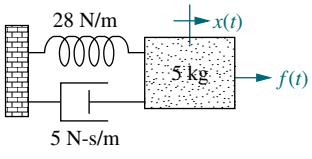


FIGURE P4.7

26. For the system shown in Figure P4.8, a step torque is applied at $\theta_1(t)$. Find

a. The transfer function, $G(s) = \theta_2(s)/T(s)$.

b. The percent overshoot, settling time, and peak time for $\theta_2(t)$. [Section: 4.6]

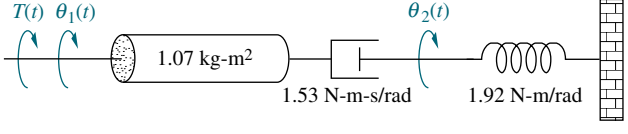


FIGURE P4.8

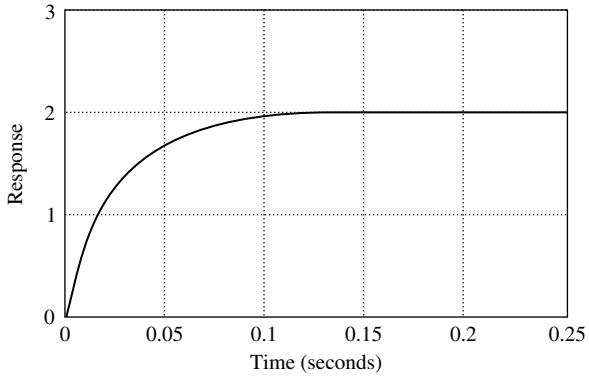
27. Derive the unit step response for each transfer function in Example 4.8. [Section: 4.7].

28. Find the percent overshoot, settling time, rise time, and peak time for

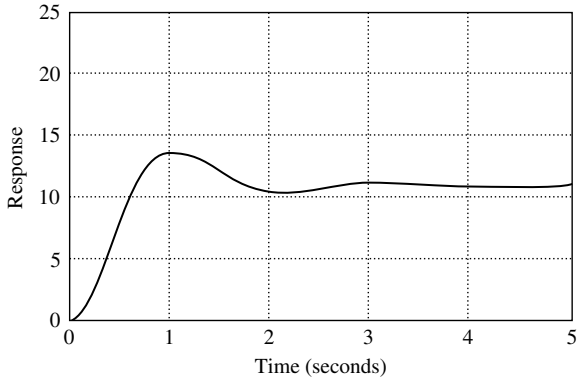
$$T(s) = \frac{14.145}{(s^2 + 0.842s + 2.829)(s + 5)}$$ [Section: 4.7]

29. For each of the unit step responses shown in Figure P4.9, find the transfer function of the system. [Sections: 4.3, 4.6].

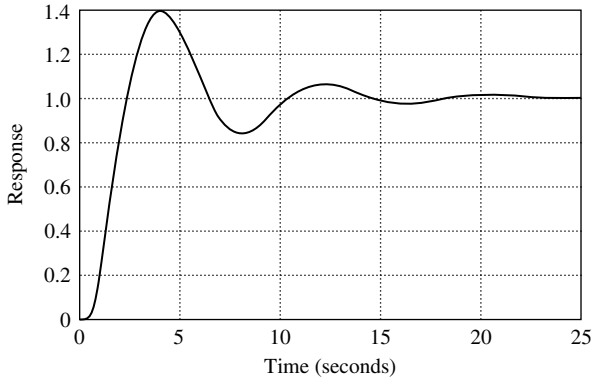
WileyPLUS WPCS Control Solutions



(*a*)



(*b*)



(*c*)

FIGURE P4.9

30. For the following response functions, determine if pole-zero cancellation can be approximated. If it can, find percent overshoot, settling time, rise time, and peak time. [Section: 4.8].

a. $C(s) = \dfrac{(s+3)}{s(s+2)(s^2+3s+10)}$

b. $C(s) = \dfrac{(s+2.5)}{s(s+2)(s^2+4s+20)}$

c. $C(s) = \dfrac{(s+2.1)}{s(s+2)(s^2+s+5)}$

d. $C(s) = \dfrac{(s+2.01)}{s(s+2)(s^2+5s+20)}$

31. Using MATLAB, plot the time response of Problem 30**a** and from the plot determine percent overshoot, settling time, rise time, and peak time. [Section: 4.8]

MATLAB ML

32. Find peak time, settling time, and percent overshoot for only those responses below that can be approximated as second-order responses. [Section: 4.8].

a. $c(t) = 0.003500 - 0.001524e^{-4t} - 0.001976e^{-3t}\cos(22.16t) - 0.0005427e^{-3t}\sin(22.16t)$

b. $c(t) = 0.05100 - 0.007353e^{-8t} - 0.007647e^{-6t}\cos(8t) - 0.01309e^{-6t}\sin(8t)$

c. $c(t) = 0.009804 - 0.0001857e^{-5.1t} - 0.009990e^{-2t}\cos(9.796t) - 0.001942e^{-2t}\sin(9.796t)$

d. $c(t) = 0.007000 - 0.001667e^{-10t} - 0.008667e^{-2t}\cos(9.951t) - 0.0008040e^{-2t}\sin(9.951t)$

33. For each of the following transfer functions with zeros, find the component parts of the unit step response: (1) the derivative of the response without a zero and (2) the response without a zero, scaled to the negative of the zero value. Also, find and plot the total response. Describe any nonminimum-phase behavior. [Section: 4.8].

a. $G(s) = \dfrac{s+2}{s^2+3s+36}$

b. $G(s) = \dfrac{s-2}{s^2+3s+36}$

34. Use MATLAB's Simulink to obtain the step response of a system,

Simulink SL

$$G(s) = \frac{1}{s^2+3s+10}$$

under the following conditions: [Section: 4.9]

a. The system is linear and driven by an amplifier whose gain is 10.

b. An amplifier whose gain is 10 drives the system. The amplifier saturates at ±0.25 volts. Describe the effect of the saturation on the system's output.

c. An amplifier whose gain is 10 drives the system. The amplifier saturates at ±0.25 volts. The system drives a 1:1 gear train that has backlash. The deadband width of the backlash is 0.02 rad. Describe the effect of saturation and backlash on the system's output.

35. A system is represented by the state and output equations that follow. Without solving the state equation, find the poles of the system. [Section: 4.10]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -2 & -1 \\ -3 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}u(t)$$

$$y = [3 \quad 2]\mathbf{x}$$

36. A system is represented by the state and output equations that follow. Without solving the state equation, find [Section: 4.10]

WileyPLUS WPCS Control Solutions

State Space SS

a. the characteristic equation;

b. the poles of the system

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 2 & 3 \\ 0 & 6 & 5 \\ 1 & 4 & 2 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}u(t)$$

$$y = [1 \quad 2 \quad 0]\mathbf{x}$$

37. Given the following state-space representation of a system, find $Y(s)$: [Section: 4.10]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & 2 \\ -3 & -1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}sin\,3t$$

$$y = [1 \quad 2]\mathbf{x}; \quad \mathbf{x}(0) = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

38. Given the following system represented in state space, solve for $Y(s)$ using the

State Space SS

Laplace transform method for solution of the state equation: [Section: 4.10]

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -4 & 1 \\ 0 & 0 & -6 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e^{-t}$$

$$y = [0 \quad 0 \quad 1]\mathbf{x}; \ \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

39. Solve the following state equation and output equation for $y(t)$, where $u(t)$ is the unit step. Use the Laplace transform method. [Section: 4.10]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -2 & 0 \\ -1 & -1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

$$y = [0 \quad 1]\mathbf{x}; \ \mathbf{x}(0) = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

40. Solve for $y(t)$ for the following system represented in state space, where $u(t)$ is the unit step. Use the Laplace transform approach to solve the state equation. [Section: 4.10]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -3 & 1 & 0 \\ 0 & -6 & 1 \\ 0 & 0 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} u(t)$$

$$y = [0 \ 1 \ 1]\mathbf{x}; \ \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

41. Use MATLAB to plot the step response of Problem 40. [Section: 4.10]

MATLAB ML

42. Repeat Problem 40 using MATLAB's Symbolic Math Toolbox and Eq. (4.96). In addition, run your program with an initial condition,

Symbolic Math SM

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \text{ [Section : 4.10]}$$

43. Using classical (not Laplace) methods only, solve for the state-transition matrix, the state vector, and the output of the system represented here: [Section: 4.11]

WileyPLUS WPCS Control Solutions

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & -5 \end{bmatrix}\mathbf{x}; \ y = [1 \ \ 2]\mathbf{x};$$

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

44. Using classical (not Laplace) methods only, solve for the state-transition matrix, the state vector, and the output of the system represented here, where $u(t)$ is the unit step: [Section: 4.11].

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$y = [3 \quad 4]\mathbf{x}; \ \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

45. Solve for $y(t)$ for the following system represented in state space, where $u(t)$ is the unit step. Use the classical approach to solve the state equation. [Section: 4.11]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -2 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -6 & -1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y = [1 \ 0 \ 0]\mathbf{x}; \ \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

46. Repeat Problem 45 using MATLAB's Symbolic Math Toolbox and Eq. (4.109). In addition, run your program with an initial condition,

Symbolic Math SM

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \text{ [Section : 4.11]}$$

47. Using methods described in Appendix H.1 located at www.wiley.com/college/nise simulate the following system and plot the step response. Verify the expected values of percent overshoot, peak time, and settling time.

State Space SS

$$T(s) = \frac{1}{s^2 + 0.8s + 1}$$

48. Using methods described in Appendix H.1 located at www.wiley.com/college/nise, simulate the following system and plot the output, $y(t)$, for a step input:

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -10 & -7 & 1 \\ 0 & 0 & -2 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = [1 \;\; 1 \;\; 0]\mathbf{x}; \;\; \mathbf{x}(0) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

49. A human responds to a visual cue with a physical response, as shown in Figure P4.10. The transfer function that relates the output physical response, $P(s)$, to the input visual command, $V(s)$, is

$$G(s) = \frac{P(s)}{V(s)} = \frac{(s+0.5)}{(s+2)(s+5)}$$

(*Stefani, 1973*). Do the following:

a. Evaluate the output response for a unit step input using the Laplace transform.

b. Represent the transfer function in state space.

State Space SS

c. Use MATLAB to simulate the system and obtain a plot of the step response.

MATLAB ML

50. Industrial robots are used for myriad applications. Figure P4.11 shows a robot used to move 55-pound bags of salt pellets; a vacuum head lifts the bags before positioning. The robot can move as many as 12 bags per minute (*Schneider, 1992*). Assume a model for the open-loop swivel controller and plant of



FIGURE P4.11 Vacuum robot lifts two bags of salt.

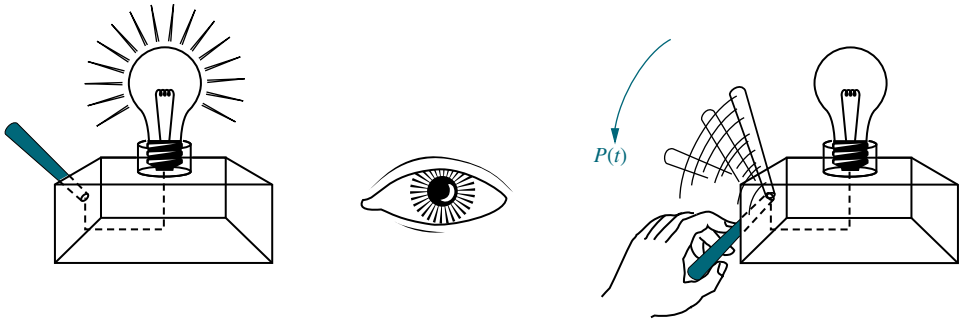
$$G_e(s) = \frac{\omega_o(s)}{V_i(s)} = \frac{K}{(s+10)(s^2+4s+10)}$$

where $\omega_o(s)$ is the Laplace transform of the robot's output swivel velocity and $V_i(s)$ is the voltage applied to the controller.

a. Evaluate percent overshoot, settling time, peak time, and rise time of the response of the open-loop swivel velocity to a step-voltage input. Justify all second-order assumptions.

b. Represent the open-loop system in state space.

State Space SS



Step 1: Light source on Step 2: Recognize light source Step 3: Respond to light source

FIGURE P4.10 Steps in determining the transfer function relating output physical response to the input visual command

c. Use MATLAB or any other computer program to simulate the system and compare your results to **a**. MATLAB ML

51. Anesthesia induces muscle relaxation (paralysis) and unconsciousness in the patient. Muscle relaxation can be monitored using electromyogram signals from nerves in the hand; unconsciousness can be monitored using the cardiovascular system's mean arterial pressure. The anesthetic drug is a mixture of isoflurane and atracurium. An approximate model relating muscle relaxation to the percent isoflurane in the mixture is

$$\frac{P(s)}{U(s)} = \frac{7.63 \times 10^{-2}}{s^2 + 1.15s + 0.28}$$

where $P(s)$ is muscle relaxation measured as a fraction of total paralysis (normalized to unity) and $U(s)$ is the percent mixture of isoflurane (*Linkens*, *1992*). [Section: 4.6]

a. Find the damping ratio and the natural frequency of the paralysis transient response.

b. Find the maximum possible percent paralysis if a 2% mixture of isoflurane is used.

c. Plot the step response of paralysis if a 1% mixture of isoflurane is used.

d. What percent isoflurane would have to be used for 100% paralysis?

52. To treat acute asthma, the drug theophylline is infused intravenously. The rate of change of the drug concentration in the blood is equal to the difference between the infused concentration and the eliminated concentration. The infused concentration is $i(t)/V_d$, where $i(t)$ is the rate of flow of the drug by weight and V_d is the apparent volume and depends on the patient. The eliminated concentration is given by $k_{10}c(t)$, where $c(t)$ is the current concentration of the drug in the blood and k_{10} is the elimination rate constant. The theophylline concentration in the blood is critical—if it is too low, the drug is ineffective; if too high, the drug is toxic (*Jannett*, *1992*). You will help the doctor with your calculations.

a. Derive an equation relating the desired blood concentration, C_D, to the required infusion rate by weight of the drug, I_R.

b. Derive an equation that will tell how long the drug must be administered to reach the desired blood concentration. Use both rise time and settling time.

c. Find the infusion rate of theophylline if $V_D = 600$ ml, $k_{10} = 0.07\ \text{h}^{-1}$, and the required blood level of the drug is 12 mcg/ml ("mcg" means micrograms). See (*Jannett, 1992*); for a description of parameter values.

d. Find the rise and settling times for the constants in **c**.

53. Upper motor neuron disorder patients can benefit and regain useful function through the use of functional neuroprostheses. The design requires a good understanding of muscle dynamics. In an experiment to determine muscle responses, the identified transfer function was (*Zhou*, *1995*)

$$M(s) = \frac{2.5e^{-0.008s}(1 + 0.172s)(1 + 0.008s)}{(1 + 0.07s)^2(1 + 0.05s)^2}$$

Find the unit step response of this transfer function.

54. When electrodes are attached to the mastoid bones (right behind the ears) and current pulses are applied, a person will sway forward and backward. It has been found that the transfer function from the current to the subject's angle (in degrees) with respect to the vertical is given by (*Nashner*, *1974*)

$$\frac{\theta(s)}{I(s)} = \frac{5.8(0.3s + 1)e^{-0.1s}}{(s + 1)(s^2/1.2^2 + 0.6s/1.2 + 1)}$$

a. Determine whether a dominant pole approximation can be applied to this transfer function.

b. Find the body sway caused by a 250 μA pulse of 150 msec duration.

55. A MOEMS (optical MEMS) is a MEMS (Micro Electromechanical Systems) with an optical fiber channel that takes light generated from a laser diode. It also has a photodetector that measures light intensity variations and outputs voltage variations proportional to small mechanical device deflections. Additionally, a voltage input is capable of deflecting the device. The apparatus can be used as an optical switch or as a variable optical attenuator, and it does not exceed 2000 μm in any dimension. Figure P4.12 shows input-output signal pairs used to identify the parameters of the system. Assume a second-order transfer function and find the system's transfer function (*Borovic*, *2005*).

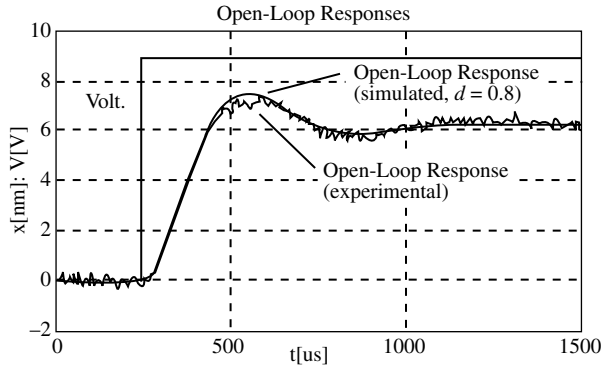


FIGURE P4.12

56. The response of the deflection of a fluid-filled catheter to changes in pressure can be modeled using a second-order model. Knowledge of the parameters of the model is important because in cardiovascular applications the undamped natural frequency should be close to five times the heart rate. However, due to sterility and other considerations, measurement of the parameters is difficult. A method to obtain transfer functions using measurements of the amplitudes of two consecutive peaks of the response and their timing has been developed (*Glantz*, *1979*). Assume that Figure P4.13 is obtained from catheter measurements. Using the information shown and assuming a second-order model excited by a unit step input, find the corresponding transfer function.

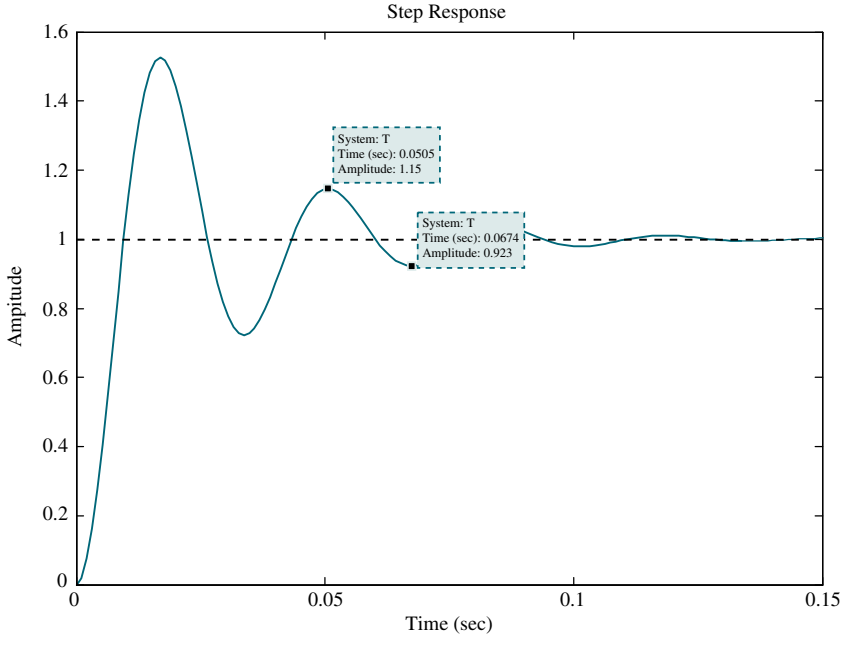


FIGURE P4.13

57. Several factors affect the workings of the kidneys. For example, Figure P4.14 shows how a step change in arterial flow pressure affects renal blood flow in rats. In the "hot tail" part of the experiment, peripheral thermal receptor stimulation is achieved by inserting the rat's tail in heated water. Variations between different test subjects are indicated by the vertical lines. It has been argued that the "control" and "hot tail" responses are identical except for their steady-state values (*DiBona*, *2005*).

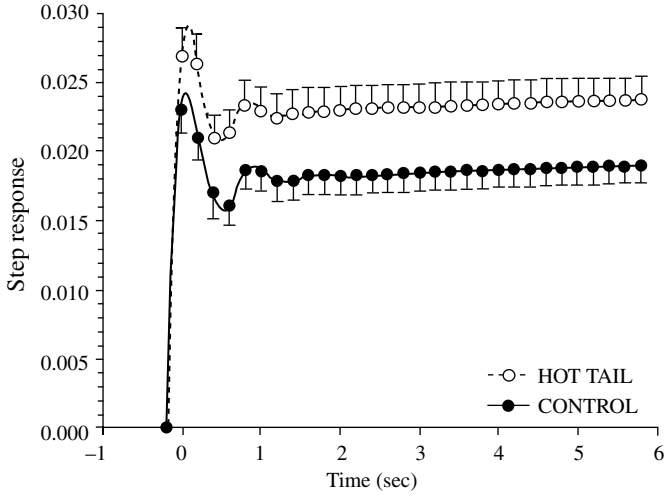


FIGURE P4.14

a. Using Figure P4.14, obtain the normalized ($c_{\text{final}} = 1$) transfer functions for both responses.

b. Use MATLAB to prove or disprove the assertion about the "control" and "hot tail" responses. MATLAB ML

58. The transfer function of a nano-positioning device capable of translating biological samples within a few μm uses a piezoelectric actuator and a linear variable differential transformer (LDVT) as a displacement sensor. The transfer function from input to displacement has been found to be (*Salapaka*, *2002*)

$$G(s) = \frac{9.7 \times 10^4 (s^2 - 14400s + 106.6 \times 10^6)}{(s^2 + 3800s + 23.86 \times 10^6)(s^2 + 240s + 2324.8 \times 10^3)}$$

Use a dominant-pole argument to find an equivalent transfer function with the same numerator but only three poles. MATLAB ML Use MATLAB to find the actual size and approximate system unit step responses, plotting them on the same graph. Explain the differences between both responses given that both pairs of poles are so far apart.

59. At some point in their lives most people will suffer from at least one onset of low back pain. This disorder can trigger excruciating pain and temporary disability, but its causes are hard to diagnose. It is well known that low back pain alters motor trunk patterns; thus it is of interest to study the causes for these alterations and their extent. Due to the different possible causes of this type of pain, a "control" group of people is hard to obtain for laboratory studies. However, pain can be stimulated in healthy people and muscle movement ranges can be compared. Controlled back pain can be induced by injecting saline solution directly into related muscles or ligaments. The transfer function from infusion rate to pain response was obtained experimentally by injecting a 5% saline solution at six different infusion rates over a period of 12 minutes. Subjects verbally rated their pain every 15 seconds on a scale from 0 to 10, with 0 indicating no pain and 10 unbearable pain. Several trials were averaged and the data was fitted to the following transfer function:

$$G(s) = \frac{9.72 \times 10^{-8}(s + 0.0001)}{(s + 0.009)^2(s^2 + 0.018s + 0.0001)}$$

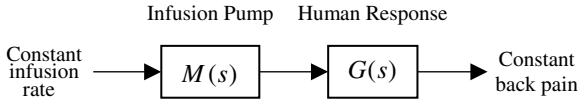


FIGURE P4.15

For experimentation it is desired to build an automatic dispensing system to make the pain level constant as shown in Figure P4.15. It follows that ideally the injection system transfer function has to be

$$M(s) = \frac{1}{G(s)}$$

to obtain an overall transfer function $M(s)G(s) \approx 1$. However, for implementation purposes $M(s)$ must have at least one more pole than zeros (*Zedka*, *1999*). Find a suitable transfer function, $M(s)$ by inverting $G(s)$ and adding poles that are far from the imaginary axis.

60. An artificial heart works in closed loop by varying its pumping rate according to changes in signals from the recipient's nervous system. For feedback compensation design it is important to know the heart's open-loop transfer function. To identify this transfer function, an artificial heart is implanted in a calf while the main parts of the original heart are left in place. Then the atrial pumping rate in the original heart is measured while step input changes are effected on the artificial heart. It has been found that, in general, the obtained response closely resembles that of a second-order system. In one such experiment it was found that the step response has a %OS = 30% and a time of first peak T_p = 127 sec (*Nakamura*, *2002*). Find the corresponding transfer function.

61. An observed transfer function from voltage potential to force in skeletal muscles is given by (*Ionescu*, *2005*)

$$T(s) = \frac{450}{(s + 5)(s + 20)}$$

a. Obtain the system's impulse response.

b. Integrate the impulse response to find the step response.

c. Verify the result in Part **b** by obtaining the step response using Laplace transform techniques.

62. In typical conventional aircraft, longitudinal flight model linearization results in transfer functions with

two pairs of complex conjugate poles. Consequently, the natural response for these airplanes has two modes in their natural response. The "short period" mode is relatively well-damped and has a high-frequency oscillation. The "plugoid mode" is lightly damped and its oscillation frequency is relatively low. For example, in a specific aircraft the transfer function from wing elevator deflection to nose angle (angle of attack) is (*McRuer, 1973*)

$$\frac{\theta(s)}{\delta_e(s)} = -\frac{26.12(s+0.0098)(s+1.371)}{(s^2+8.99\times10^{-3}s+3.97\times10^{-3})(s^2+4.21s+18.23)}$$

a. Find which of the poles correspond to the short period mode and which to the phugoid mode.

b. Peform a "phugoid approximation" (dominant-pole approximation), retaining the two poles and the zero closest to the; ω-axis.

c. Use MATLAB to compare the step responses of the original transfer function and the approximation.

MATLAB ML

63. A crosslapper is a machine that takes as an input a light fiber fabric and produces a heavier fabric by laying the original fabric in layers rotated by 90 degrees. A feedback system is required in order to maintain consistent product width and thickness by controlling its carriage velocity. The transfer function from servomotor torque, T_m(s), to carriage velocity, $Y(s)$, was developed for such a machine (*Kuo, 2008*). Assume that the transfer function is:

MATLAB ML

$$G(s) = \frac{Y(s)}{T_m(s)} = \frac{33s^4+202s^3+10061s^2+24332s+170704}{s^7+8s^6+464s^5+2411s^4+52899s^3+167829s^2+913599s+1076555}$$

a. Use MATLAB to find the partial fraction residues and poles of $G(s)$.

b. Find an approximation to $G(s)$ by neglecting the second-order terms found in **a**.

c. Use MATLAB to plot on one graph the step response of the transfer function given above and the approximation found in **b**. Explain the differences between the two plots.

64. Although the use of fractional calculus in control systems is not new, in the last decade there is increased interest in its use for several reasons. The most relevant are that fractional calculus differential equations may model certain systems with higher accuracy than integer differential equations, and that fractional calculus compensators might exhibit advantageous properties for control system design. An example of a transfer function obtained through fractional calculus is:

MATLAB ML

$$G(s) = \frac{1}{s^{2.5}+4s^{1.7}+3s^{0.5}+5}$$

This function can be approximated with an integer rational transfer function (integer powers of s) using Oustaloup's method (*Xue, 2005*). We ask you now to do a little research and consult the aforementioned reference to find and run an M-file that will calculate the integer rational transfer function approximation to $G(s)$ and plot its step response.

65. Mathematical modeling and control of pH processes are quite challenging since the processes are highly nonlinear, due to the logarithmic relationship between the concentration of hydrogen ions [H+] and pH level. The transfer function from input pH to output pH is $G_a(s) = \frac{Y_a(s)}{X_a(s)} = \frac{14.49e^{-4s}}{1478.26s+1}$. $G_a(s)$ is a model for the anaerobic process in a wastewater treatment system in which methane bacteria need the pH to be maintained in its optimal range from 6.8 to 7.2 (*Jiayu, 2009*). Similarly, (*Elarafi, 2008*) used empirical techniques to model a pH neutralization plant as a second-order system with a pure delay, yielding the following transfer function relating output pH to input pH:

$$G_p(s) = \frac{Y_p(s)}{X_p(s)} = \frac{1.716\times10^{-5}e^{-30s}}{s^2+6.989\times10^{-3}s+1.185\times10^{-6}}$$

a. Find analytical expressions for the unit-step responses $y_a(t)$ and $y_p(t)$ for the two

processes, $G_a(s)$ and $G_p(s)$.
(Hint: Use the time shift theorem in Table 2.2).

b. Use Simulink to plot $y_a(t)$ and $y_p(t)$ on a single graph. Simulink SL

66. Using wind tunnel tests, insect flight dynamics can be studied in a very similar fashion to that of man-made aircraft. Linearized longitudinal flight equations for a bumblebee have been found in the unforced case to be

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -8.792 \times 10^{-3} & 0.56 \times 10^{-3} & -1.0 \times 10^{-3} & -13.79 \times 10^{-3} \\ -0.347 \times 10^{-3} & -11.7 \times 10^{-3} & -0.347 \times 10^{-3} & 0 \\ 0.261 & -20.8 \times 10^{-3} & -96.6 \times 10^{-3} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix}$$

where u = forward velocity; w = vertical velocity, q = angular pitch rate at center of mass, and θ = pitch angle between the flight direction and the horizontal (*Sun*, *2005*).

a. Use MATLAB to obtain the system's eigenvalues. MATLAB ML

b. Write the general form of the state-transition matrix. How many constants would have to be found?

67. A dc-dc converter is a device that takes as an input an unregulated dc voltage and provides a regulated dc voltage as its output. The output voltage may be lower (buck converter), higher (boost converter), or the same as the input voltage. Switching dc-dc converters have a semiconductor active switch (BJT or FET) that is closed periodically with a duty cycle d in a pulse width modulated (PWM) manner. For a boost converter, averaging techniques can be used to arrive at the following state equations (*Van Dijk*, *1995*): State Space SS

$$L\frac{di_L}{dt} = -(1-d)u_c + E_s$$

$$C\frac{du_C}{dt} = (1-d)i_L - \frac{u_C}{R}$$

where L and C are respectively the values of internal inductance and capacitance; i_L is the current through the internal inductor; R is the resistive load connected to the converter; E_s is the dc input voltage; and the capacitor voltage, u_C, is the converter's output.

a. Write the converter's equations in the form

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\mathbf{y} = \mathbf{Cx}$$

assuming d is a constant.

b. Using the **A**, **B**, and **C** matrices of Part **a**, obtain the converter's transfer function $\frac{U_C(s)}{E_s(s)}$.

68. An IPMC (ionic polymer-metal composite) is a Nafion sheet plated with gold on both sides. An IPMC bends when an electric field is applied across its thickness. IPMCs have been used as robotic actuators in several applications and as active catheters in biomedical applications. With the aim of improving actuator settling times, a state-space model has been developed for a 20 mm × 10 mm × 0.2 mm polymer sample (*Mallavarapu*, *2001*): State Space SS

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -8.34 & -2.26 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 12.54 & 2.26 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where u is the applied input voltage and y is the deflection at one of the material's tips when the sample is tested in a cantilever arrangement.

a. Find the state-transition matrix for the system.

b. From Eq. (4.109) in the text, it follows that if a system has zero initial conditions the system output for any input can be directly calculated from the state-space representation and the state-transition matrix using

$$y(t) = \mathbf{Cx}(t) = \int \mathbf{C}\Phi(t-\tau)\,\mathbf{B}u(\tau)d\tau$$

Use this equation to find the zero initial condition unit step response of the IPMC material sample.

c. Use MATLAB to verify that your step response calculation in Part b is correct. MATLAB ML

DESIGN PROBLEMS

69. Find an equation that relates 2% settling time to the value of f_v for the translational mechanical system shown in Figure P4.16. Neglect the mass of all components. [Section: 4.6] WileyPLUS WPCS Control Solutions

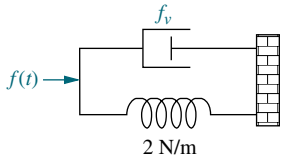


FIGURE P4.16

70. Consider the translational mechanical system shown in Figure P4.17. A 1-pound force, $f(t)$, is applied at $t = 0$. If $f_v = 1$, find K and M such that the response is characterized by a 4-second settling time and a 1-second peak time. Also, what is the resulting percent overshoot? [Section: 4.6]

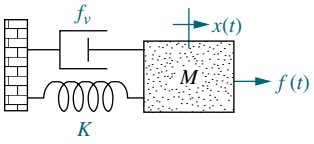


FIGURE P4.17

71. Given the translational mechanical system of Figure P4.17, where $K = 1$ and $f(t)$ is a unit step, find the values of M and f_v to yield a response with 17% overshoot and a settling time of 10 seconds. [Section: 4.6]

72. Find J and K in the rotational system shown in Figure P4.18 to yield a 30% overshoot and a settling time of 3 seconds for a step input in torque. [Section: 4.6]

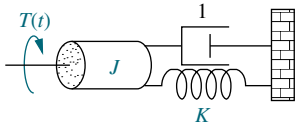


FIGURE P4.18

73. Given the system shown in Figure P4.19, find the damping, D, to yield a 30% overshoot in output angular displacement for a step input in torque. [Section: 4.6]

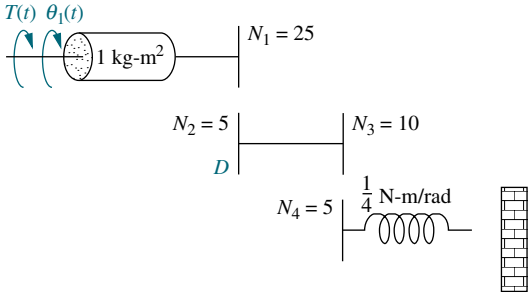


FIGURE P4.19

74. For the system shown in Figure P4.20, find N_1/N_2 so that the settling time for a step torque input is 16 seconds. [Section: 4.6]

WileyPLUS WPCS Control Solutions

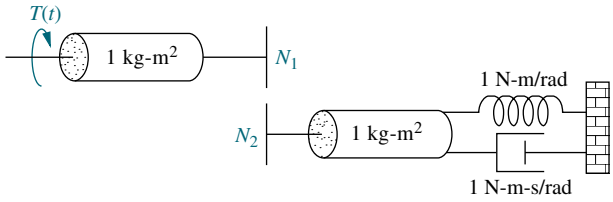


FIGURE P4.20

75. Find M and K, shown in the system of Figure P4.21, to yield $x(t)$ with 10% overshoot and 15 seconds settling time for a step input in motor torque, $T_m(t)$. [Section: 4.6]

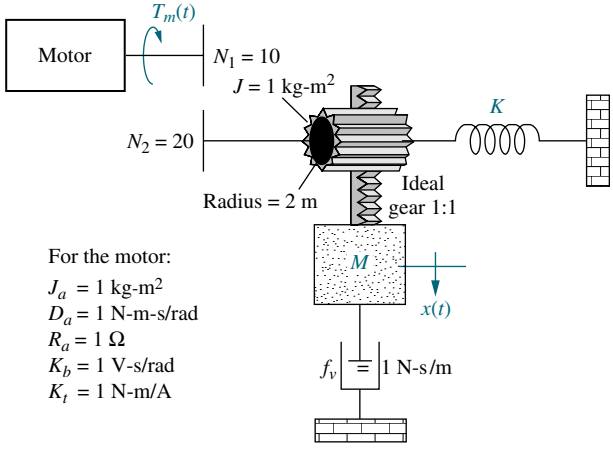


FIGURE P4.21

76. If $v_i(t)$ is a step voltage in the network shown in Figure P4.22, find the value of the resistor such that a 20% overshoot in voltage will be seen across the capacitor if $C = 10^{-6}$ F and $L = 1$ H. [Section: 4.6]

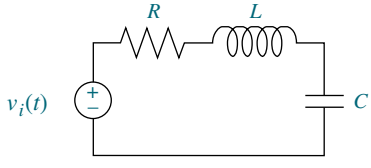


FIGURE P4.22

77. If $v_i(t)$ is a step voltage in the network shown in Figure P4.22, find the values of R and C to yield a 20% overshoot and a 1 ms settling time for $v_c(t)$ if $L = 1$ H. [Section: 4.6]

WileyPLUS WPCS Control Solutions

78. Given the circuit of Figure P4.22, where $C = 10\mu$F, find R and L to yield 15% overshoot with a settling time of 7 ms for the capacitor voltage. The input, $v(t)$, is a unit step. [Section: 4.6]

79. For the circuit shown in Figure P4.23, find the values of R_2 and C to yield 8% overshoot with a settling time of 1 ms for the voltage across the capacitor, with $v_i(t)$ as a step input. [Section: 4.6]

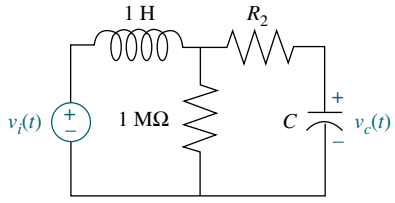


FIGURE P4.23

80. Hydraulic pumps are used as inputs to hydraulic circuits to supply pressure, just as voltage sources supply potential to electric circuits. Applications for hydraulic circuits can be found in the robotics and aircraft industries, where hydraulic actuators are used to move component parts. Figure P4.24 shows the internal parts of the pump. A barrel containing equally spaced pistons rotates about the i-axis. A swashplate, set at an angle, causes the slippers at the ends of the pistons to move the pistons in and out. When the pistons are moving across the intake port, they are extending, and when they are moving across the discharge port, they are retracting and pushing fluid from the port. The large and small actuators at the top and bottom, respectively, control the angle of the swashplate, α. The swashplate angle affects the piston stroke length. Thus, by controlling the swashplate angle, the pump discharge flow rate can be regulated. Assume the state equation for the hydraulic pump is

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} (3.45 - 14000K_c) & -0.255 \times 10^{-9} \\ 0.499 \times 10^{11} & -3.68 \end{bmatrix} \mathbf{x} + \begin{bmatrix} -3.45 + 14000K_c \\ -0.499 \times 10^{11} \end{bmatrix} \alpha_0,$$

where $\mathbf{x} = \begin{bmatrix} \alpha \\ P_d \end{bmatrix}$

and P_d is the pump discharge pressure (*Manring, 1996*). Find the value of controller flow gain, K_c, so that the damping ratio of the system's poles is 0.9.

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

81. High-speed rail pantograph. Problem 67**c** in Chapter 2 asked you to find $G(s) = (Y_h(s) - Y_{cat}(s))/F_{up}(s)$ (*O'Connor, 1997*).

a. Use the dominant poles from this transfer function and estimate percent overshoot, damping ratio, natural frequency, settling time, peak time, and rise time.

b. Determine if the second-order approximation is valid.

c. `Obtain the step response of G(s) and compare the results to Part a.`

MATLAB ML

82. Control of HIV/AIDS. In Chapter 3, Problem 31, we developed a linearized

State Space SS

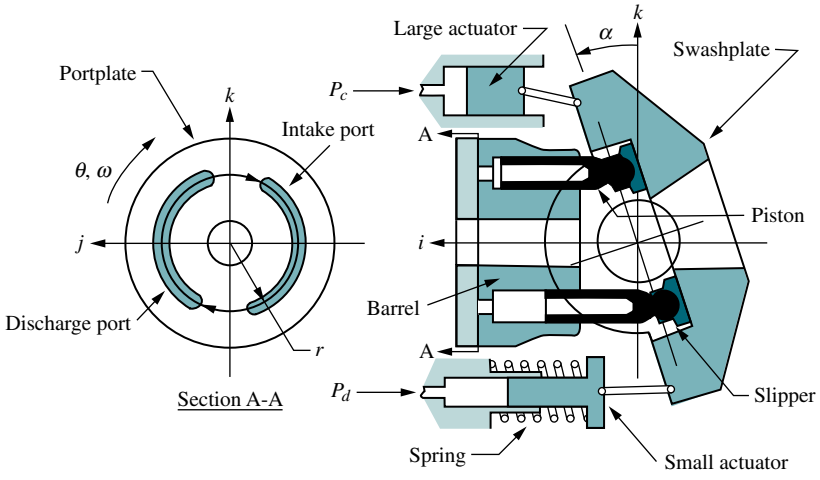


FIGURE P4.24 Pump diagram (Reprinted with permission of ASME.)

state-space model of HIV infection. The model assumed that two different drugs were used to combat the spread of the HIV virus. Since this book focuses on single-input, single-output systems, only one of the two drugs will be considered. We will assume that only RTIs are used as an input. Thus, in the equations of Chapter 3, Problem 31, $u_2 = 0$ (*Craig, 2004*).

a. Show that when using only RTIs in the linearized system of Problem 31 and substituting the typical parameter values given in the table of Problem 31**c**, the resulting state-space representation for the system is given by

$$\begin{bmatrix} \dot{T} \\ \dot{T}^* \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -0.04167 & 0 & -0.0058 \\ 0.0217 & -0.24 & 0.0058 \\ 0 & 100 & -2.4 \end{bmatrix} \times \begin{bmatrix} T \\ T^* \\ v \end{bmatrix} + \begin{bmatrix} 5.2 \\ -5.2 \\ 0 \end{bmatrix} u_1$$

$$y = [0 \quad 0 \quad 1] \begin{bmatrix} T \\ T^* \\ v \end{bmatrix}$$

b. Obtain the transfer function from RTI efficiency to virus count; namely find $\dfrac{Y(s)}{U_1(s)}$.

c. Assuming RTIs are 100% effective, what will be the steady-state change of virus count in a given infected patient? Express your answer in virus copies per ml of plasma. Approximately how much time will the medicine take to reach its maximum possible effectiveness?

83. Hybrid vehicle. Assume that the car motive dynamics for a hybrid electric vehicle (HEV) can be described by the transfer function

$$\frac{\Delta V(s)}{\Delta F_e(s)} = \frac{1}{1908s + 10}$$

where AV is the change of velocity in m/sec and ΔF_e is the change in excess motive force in N necessary to propel the vehicle.

a. Find an analytical expression for $\Delta v_{(t)}$ for a step change in excess motive force $\Delta F_e = 2650$ N.

b. `Simulate the system using MATLAB. Plot the expression found in Part a together with your simulated plot.` MATLAB ML

Cyber Exploration Laboratory

Experiment 4.1

Objective To evaluate the effect of pole and zero location upon the time response of first- and second-order systems.

Minimum Required Software Packages MATLAB, Simulink, and the Control System Toolbox

Prelab

1. Given the transfer function $G(s) = \dfrac{a}{s + a}$, evaluate settling time and rise time for the following values of a: 1, 2, 3, 4. Also, plot the poles.

2. Given the transfer function $G(s) = \dfrac{b}{s^2 + as + b}$:

a. Evaluate percent overshoot, settling time, peak time, and rise time for the following values: $a = 4$, $b = 25$. Also, plot the poles.

b. Calculate the values of a and b so that the imaginary part of the poles remains the same but the real part is increased two times over that of Prelab 2**a**, and repeat Prelab 2**a**.

c. Calculate the values of a and b so that the imaginary part of the poles remains the same but the real part is decreased $\frac{1}{2}$ time over that of Prelab 2**a**, and repeat Prelab 2**a**.

3. **a.** For the system of Prelab 2**a**, calculate the values of a and b so that the real part of the poles remains the same but the imaginary part is increased two times over that of Prelab 2**a**, and repeat Prelab 2**a**.

   b. For the system of Prelab 2**a**, calculate the values of a and b so that the real part of the poles remains the same but the imaginary part is increased four times over that of Prelab 2**a**, and repeat Prelab 2**a**.

4. **a.** For the system of Prelab 2**a**, calculate the values of a and b so that the damping ratio remains the same but the natural frequency is increased two times over that of Prelab 2**a**, and repeat Prelab 2**a**.

   b. For the system of Prelab 2**a**, calculate the values of a and b so that the damping ratio remains the same but the natural frequency is increased four times over that of Prelab 2**a**, and repeat Prelab 2**a**.

5. Briefly describe the effects on the time response as the poles are changed in each of Prelab 2, 3, and 4.

Lab

1. Using Simulink, set up the systems of Prelab 1 and plot the step response of each of the four transfer functions on a single graph by using the Simulink LTI Viewer. Also, record the values of settling time and rise time for each step response.
2. Using Simulink, set up the systems of Prelab 2. Using the Simulink LTI Viewer, plot the step response of each of the three transfer functions on a single graph. Also, record the values of percent overshoot, settling time, peak time, and rise time for each step response.
3. Using Simulink, set up the systems of Prelab 2**a** and Prelab 3. Using the Simulink LTI Viewer, plot the step response of each of the three transfer functions on a single graph. Also, record the values of percent overshoot, settling time, peak time, and rise time for each step response.
4. Using Simulink, set up the systems of Prelab 2**a** and Prelab 4. Using the Simulink LTI Viewer, plot the step response of each of the three transfer functions on a single graph. Also, record the values of percent overshoot, settling time, peak time, and rise time for each step response.

Postlab

1. For the first-order systems, make a table of calculated and experimental values of settling time, rise time, and pole location.
2. For the second-order systems of Prelab 2, make a table of calculated and experimental values of percent overshoot, settling time, peak time, rise time, and pole location.
3. For the second-order systems of Prelab 2**a** and Prelab 3, make a table of calculated and experimental values of percent overshoot, settling time, peak time, rise time, and pole location.
4. For the second-order systems of Prelab 2**a** and Prelab 4, make a table of calculated and experimental values of percent overshoot, settling time, peak time, rise time, and pole location.

5. Discuss the effects of pole location upon the time response for both first- and second-order systems. Discuss any discrepancies between your calculated and experimental values.

Experiment 4.2

Objective To evaluate the effect of additional poles and zeros upon the time response of second-order systems.

Minimum Required Software Packages MATLAB, Simulink, and the Control System Toolbox

Prelab

1. **a.** Given the transfer function $G(s) = \dfrac{25}{s^2 + 4s + 25}$, evaluate the percent overshoot, settling time, peak time, and rise time. Also, plot the poles.
   b. Add a pole at -200 to the system of Prelab 1**a**. Estimate whether the transient response in Prelab 1**a** will be appreciably affected.
   c. Repeat Prelab 1**b** with the pole successively placed at -20, -10, and -2.
2. A zero is added to the system of Prelab 1**a** at -200 and then moved to -50, -20, -10, -5, and -2. List the values of zero location in the order of the greatest to the least effect upon the pure second-order transient response.
3. Given the transfer function $G(s) = \dfrac{(25b/a)(s + a)}{(s + b)(s^2 + 4s + 25)}$, let $a = 3$ and $b = 3.01$, 3.1, 3.3, 3.5, and 4.0. Which values of b will have minimal effect upon the pure second-order transient response?
4. Given the transfer function $G(s) = \dfrac{(2500b/a)(s + a)}{(s + b)(s^2 + 40s + 2500)}$, let $a = 30$ and $b = 30.01$, 30.1, 30.5, 31, 35, and 40. Which values of b will have minimal effect upon the pure second-order transient response?

Lab

1. Using Simulink, add a pole to the second-order system of Prelab 1**a** and plot the step responses of the system when the higher-order pole is nonexistent, at -200, -20, -10, and -2. Make your plots on a single graph, using the Simulink LTI Viewer. Normalize all plots to a steady-state value of unity. Record percent overshoot, settling time, peak time, and rise time for each response.
2. Using Simulink, add a zero to the second-order system of Prelab 1**a** and plot the step responses of the system when the zero is nonexistent, at -200, -50, -20, -10, -5, and -2. Make your plots on a single graph, using the Simulink LTI Viewer. Normalize all plots to a steady-state value of unity. Record percent overshoot, settling time, peak time, and rise time for each response.
3. Using Simulink and the transfer function of Prelab 3 with $a = 3$, plot the step responses of the system when the value of b is 3, 3.01, 3.1, 3.3, 3.5, and 4.0. Make your plots on a single graph using the Simulink LTI Viewer. Record percent overshoot, settling time, peak time, and rise time for each response.

4. Using Simulink and the transfer function of Prelab 4 with $a = 30$, plot the step responses of the system when the value of b is 30, 30.01, 30.1, 30.5, 31, 35, and 40. Make your plots on a single graph, using the Simulink LTI Viewer. Record percent overshoot, settling time, peak time, and rise time for each response.

Postlab

1. Discuss the effect upon the transient response of the proximity of a higher-order pole to the dominant second-order pole pair.
2. Discuss the effect upon the transient response of the proximity of a zero to the dominant second-order pole pair. Explore the relationship between the length of the vector from the zero to the dominant pole and the zero's effect upon the pure second-order step response.
3. Discuss the effect of pole-zero cancellation upon the transient response of a dominant second-order pole pair. Allude to how close the canceling pole and zero should be and the relationships of (1) the distance between them and (2) the distance between the zero and the dominant second-order poles.

Experiment 4.3

Objective To use LabVIEW Control Design and Simulation Module for time performance analysis of systems.

Minimum Required Software Packages LabVIEW with the Control Design and Simulation Module

Prelab One of the experimental direct drive robotic arms built at the MTT Artificial Intelligence Laboratory and the CMU Robotics Institute can be represented as a feedback control system with a desired angular position input for the robot's joint position and an angular position output representing the actual robot's joint position.

The forward path consists of three transfer functions in cascade; (1) a compensator, $G_c(s)$, to improve performance; (2) a power amplifier of gain, $K_a = 1$; and (3) the transfer function of the motor and load, $G(s) = 2292/s(s + 75.6)$. Assume a unity-feedback system. Initially the system will be controlled with $G_c(s) = 0.6234$, which is called a proportional controller (*McKerrow, 1991*).

1. Obtain the closed-loop system transfer function and use MATLAB to make a plot of the resulting unit step response.
2. Repeat with $G_c(s) = 3.05 + 0.04s$, which is called a PD controller.
3. Compare both responses and draw conclusions regarding their time domain specifications.

Lab Create a LabVIEW VI that uses a simulation loop to implement both controllers given in the Prelab. Plot the responses on the same graph for comparison purposes.

Postlab Compare the responses obtained using your LabVIEW VI with those obtained in the Prelab.

Experiment 4.4

Objective To use the LabVIEW Control Design and Simulation Module to evaluate the effect of pole location upon the time response of second-order systems.

Minimum Required Software Packages LabVIEW with the Control Design and Simulation Module.

Prelab Solve the Cyber Exploration Laboratory Experiment 4.1 Prelab, Part 2.

Lab Build a LabVIEW VI to implement the functions studied in the Prelab of Cyber Exploration Laboratory 4.I, Part 2.

Specifically for Prelab Part **a**, your front panel will have the coefficients of the second-order transfer function as inputs. The front panel will also have the following indicators: (1) the transfer function; (2) the state-space representation; (3) the pole locations; (4) the step response graph; (5) the time response of the two states on the same graph; (6) the time response parametric data including rise time, peak time, settling time, percent overshoot, peak value; and final value.

For Prelab, Part **b**, your front panel will also have the following indicators: (1) the step response graph, and (2) the parametric data listed above for Prelab, Part **a**, but specific to Part **b**.

For Prelab, Part **c**, your front panel will also have the following indicators: (1) the step response graph, and (2) the parametric data listed above for Prelab, Part **a**, but specific to Part **c**.

Run the VI to obtain the data from the indicators.

Postlab Use your results to discuss the effect of pole location upon the step response.

Bibliography

Borovic, B., Liu, A. Q., Popa, D., and Lewis, F. L. Open-Loop versus Closed-Loop Control of MEMS Devices: Choices and Issues. *Journal of Micromechanics, Microengineering*, vol. 15, 2005, pp. 1917–1924.

Craig, I. K., Xia, X., and Venter, J. W.; Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

DiBona, G. F. Physiology in Perspective: The Wisdom of the Body. Neural Control of the Kidney. *American Journal of Physiology–Regulatory, Integrative and Comparative Physiology*, vol. 289, 2005, pp. R633–R641.

Dorf, R. C. *Introduction to Electric Circuits*, 2d ed. Wiley, New York, 1993.

Elarafi, M. G. M. K., and Hisham, S. B. Modeling and Control of pH Neutralization Using Neural Network Predictive Controller. *International Conference on Control, Automation and Systems 2008,* Seoul, Korea. Oct. 14–17, 2008.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems*, 2d ed. Addison-Wesley, Reading, MA, 1991.

Glantz, A. S., and Tyberg, V. J. Determination of Frequency Response from Step Response: Application to Fluid-Filled Catheters. *American Journal of Physiology*, vol. 2, 1979, pp. H376–H378.

Good, M. C., Sweet, L. M., and Strobel, K. L. Dynamic Models for Control System Design of Integrated Robot and Drive Systems. *Journal of Dynamic Systems, Measurement, and Control*, March 1985, pp. 53–59.

Ionescu, C., and De Keyser, R. Adaptive Closed-Loop Strategy for Paralyzed Skeletal Muscles. Proceedings of the IASTED International Conference on Biomedical Engineering, 2005.

Jannett, T. C., and Aragula, S. Simulation of Adaptive Control of Theophylline Concentrations. *IEEE Control Systems*, December 1992, pp. 32–37.

Jiayu, K., Mengxiao W., Linan, M., and Zhongjun, X. Cascade Control of pH in an Anaerobic Waste Water Treatment System, *3d International Conference on Bioinformatics and Biomedical Engineering*, 2009.

Johnson, H. et al. *Unmanned Free-Swimming Submersible (UFSS) System Description*. NRL Memorandum Report 4393. Naval Research Laboratory, Washington, D.C. 1980.

Kuo, B. C. *Automatic Control Systems*, 5th ed. Prentice Hall, Upper Saddle River, NJ, 1987.

Kuo, C-F. J., Tsai, C-C., and Tu, H-M. Carriage Speed Control of a Cross-lapper System for Nonwoven Web Quality. *Fibers and Polymers*, vol. 9, no. 4, 2008, pp. 495–502.

Linkens, D. A. Adaptive and Intelligent Control in Anesthesia. *IEEE Control Systems*, December 1992, pp. 6–11.

Mallavarapu, K., Newbury, K., and Leo, D. J. *Feedback Control of the Bending Response of Ionic Polymer-Metal Composite Actuators.* Proceedings of the SPIE, vol. 4329, 2001, pp. 301–310.

Manring, N. D., and Johnson, R. E. Modeling and Designing a Variable-Displacement Open-Loop Pump. *Journal of Dynamic Systems, Measurements, and Control*, vol. 118, June 1996, pp. 267–271.

McKerrow, P. J. *Introduction to Robotics.* Addison-Wesley, Singapore, 1991.

McRuer, D., Ashkenas, I., and Graham, D. *Aircraft Dynamics and Automatic Control*. Princeton University Press, 1973.

Nakamura, M. et al. Transient Response of Remnant Atrial Heart Rate to Step Changes in Total Artificial Heart Output. *Journal of Artificial Organs*, vol. 5, 2002, pp. 6–12.

Nashner, L. M., and Wolfson, P. Influence of Head Position and Proprioceptive Cues on Short Latency Postural Reflexes Evoked by Galvanic Stimulation of the Human Labyrinth. *Brain Research*, vol. 67, 1974, pp. 255–268.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. l–4.

Ogata, K. *Modern Control Engineering*, 2d ed. Prentice Hall, Upper Saddle River, NJ, 1990.

Philips, C. L., and Nagle, H. T. *Digital Control Systems Analysis and Design*. Prentice Hall, Upper Saddle River, NJ, 1984.

Salapaka, S., Sebastian, A., Cleveland, J. P., and Salapaka, M. V High Bandwidth Nano-Positioner: A Robust Control Approach. *Review of Scientific Instruments*, vol. 73, No. 9, 2002, pp. 3232–3241.

Sawusch, M. R., and Summers, T. A. *1001 Things to Do with Your Macintosh*. TAB Books, Blue Ridge Summit, PA, 1984.

Schneider, R. T. Pneumatic Robots Continue to Improve. *Hydraulics & Pneumatics*, October 1992, pp. 38–39.

Stefani, R. T. *Modeling Human Response Characteristics.* COED Application Note No. 33. Computers in Education Division of ASEE, 1973.

Sun, M., and Xiong, Y. Dynamic Flight Stability of a Hovering Bumblebee. *Journal of Experimental Biology*, vol. 208, 2005, pp. 447–459.

Timothy, L. K., and Bona, B. E. *State Space Analysis: An Introduction*. McGraw-Hill, New York, 1968.

Van Dijk, E., Spruijt, J. N., O'Sullivan, D. M., and Klaasens, J. B. PWM-Switch Modeling of DC-DC Converters. *IEEE Transactions on Power Electronics*, vol. 10, 1995, pp. 659–665.

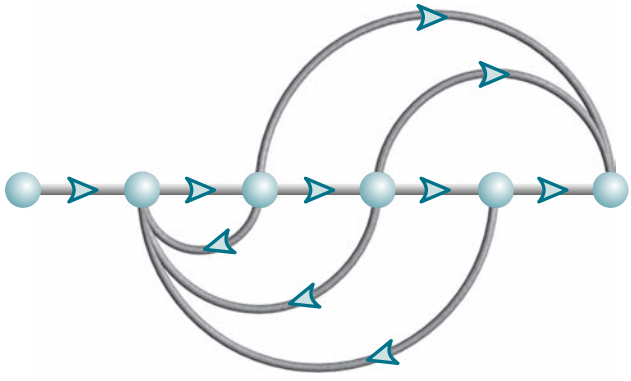
Xue, D., and Chen, Y. D. Sub-Optimum H_2 Rational Approximations to Fractional Order Linear Systems. *Proceedings of IDET/CIE 2005*. ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Long Beach. CA, 2005. pp. 1–10.

Zedka, M., Prochazka, A., Knight, B., Gillard, D., and Gauthier, M. Voluntary and Reflex Control of Human Back Muscles During Induced Pain. *Journal of Physiology*, vol. 520, 1999, pp. 591–604.

Zhou, B. H., Baratta, R. V., Solomonow, M., and D'Ambrosia, R. D. The Dynamic Response of the Cat Ankle Joint During Load-Moving Contractions. *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 4, 1995, pp. 386–393.

Reduction of Multiple Subsystems

5



Chapter Learning Outcomes

After completing this chapter the student will be able to:

- Reduce a block diagram of multiple subsystems to a single block representing the transfer function from input to output (Sections 5.1–5.2)
- Analyze and design transient response for a system consisting of multiple subsystems (Section 5.3)
- Convert block diagrams to signal-flow diagrams (Section 5.4)
- Find the transfer function of multiple subsystems using Mason's rule (Section 5.5)
- Represent state equations as signal-flow graphs (Section 5.6)
- Represent multiple subsystems in state space in cascade, parallel, controller canonical, and observer canonical forms (Section 5.7)
- Perform transformations between similar systems using transformation matrices; and diagonalize a system matrix (Section 5.8)

State Space SS

State Space SS

State Space SS

Case Study Learning Outcomes

You will be able to demonstrate your knowledge of the chapter objectives with case studies as follows:

- Given the antenna azimuth position control system shown on the front endpapers, you will be able to (a) find the closed-loop transfer function that represents the system from input to output; (b) find a state-space representation for the closed-loop system; (c) predict, for a simplified system model, the percent overshoot,

settling time, and peak time of the closed-loop system for a step input; (d) calculate the step response for the closed-loop system; and (e) for the simplified model, design the system gain to meet a transient response requirement.

State Space
SS

- Given the block diagrams for the Unmanned Free-Swimming Submersible (UFSS) vehicle's pitch and heading control systems on the back endpapers, you will be able to represent each control system in state space.

5.1 Introduction

We have been working with individual subsystems represented by a block with its input and output. More complicated systems, however, are represented by the interconnection of many subsystems. Since the response of a single transfer function can be calculated, we want to represent multiple subsystems as a single transfer function. We can then apply the analytical techniques of the previous chapters and obtain transient response information about the entire system.

In this chapter, multiple subsystems are represented in two ways: as block diagrams and as signal-flow graphs. Although neither representation is limited to a particular analysis and design technique, block diagrams are usually used for frequency-domain analysis and design, and signal-flow graphs for state-space analysis.

Signal-flow graphs represent transfer functions as lines, and signals as small-circular nodes. Summing is implicit. To show why it is convenient to use signal-flow graphs for state-space analysis and design, consider Figure 3.10. A graphical representation of a system's transfer function is as simple as Figure 3.10(a). However, a graphical representation of a system in state space requires representation of each state variable, as in Figure 3.10(b). In that example, a single-block transfer function requires seven blocks and a summing junction to show the state variables explicitly. Thus, signal-flow graphs have advantages over block diagrams, such as Figure 3.10(b): They can be drawn more quickly, they are more compact, and they emphasize the state variables.

We will develop techniques to reduce each representation to a single transfer function. Block diagram algebra will be used to reduce block diagrams and Mason's rule to reduce signal-flow graphs. Again, it must be emphasized that these methods are typically used as described. As we shall see, however, either method can be used for frequency-domain or state-space analysis and design.

5.2 Block Diagrams

As you already know, a subsystem is represented as a block with an input, an output, and a transfer function. Many systems are composed of multiple subsystems, as in Figure 5.1. When multiple subsystems are interconnected, a few more schematic elements must be added to the block diagram. These new elements are *summing junctions* and *pickoff points*. All component parts of a block diagram for a linear, time-invariant system are shown in Figure 5.2. The characteristic of the summing junction shown in Figure 5.2(c) is that the output signal, $C(s)$, is the algebraic sum of the input signals, $R_1(s)$, $R_2(s)$, and $R_3(s)$. The figure shows three inputs, but any

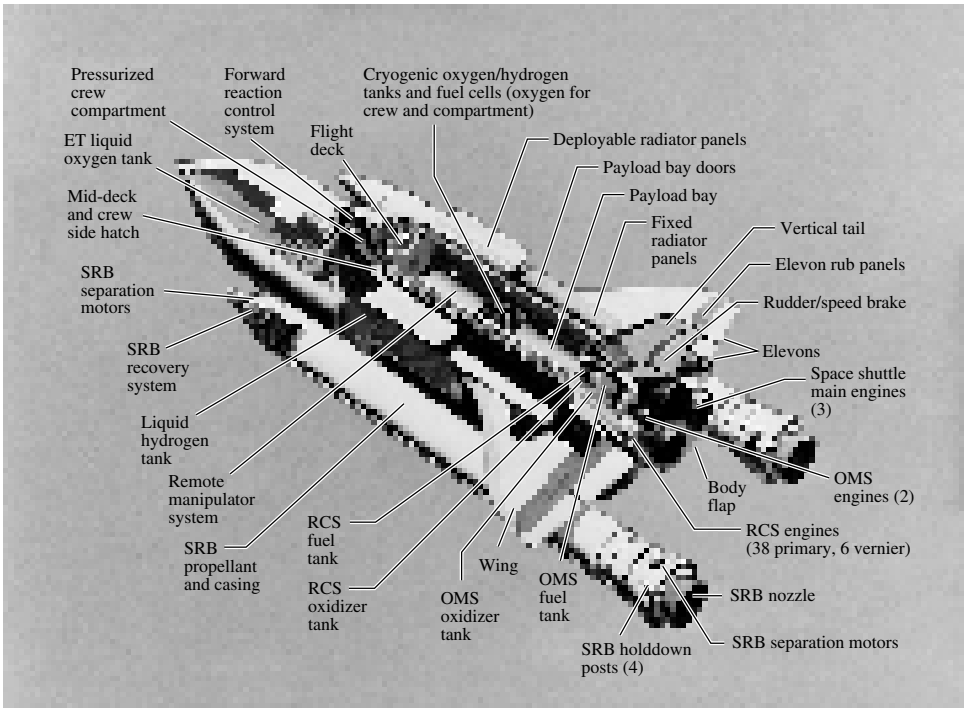


FIGURE 5.1 The space shuttle consists of multiple subsystems. Can you identify those that are control systems or parts of control systems?

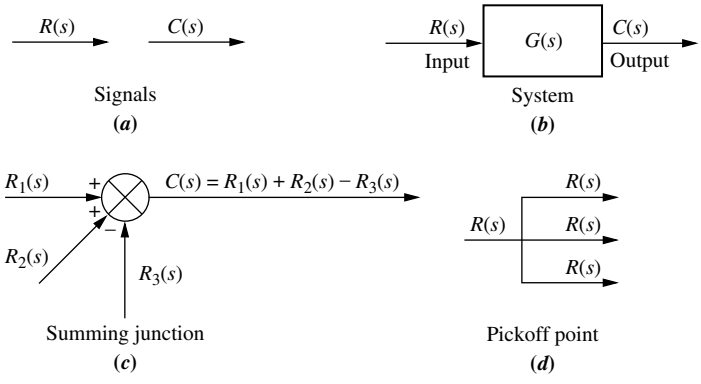


FIGURE 5.2 Components of a block diagram for a linear, time-invariant system

number can be present. A pickoff point, as shown in Figure 5.2(d), distributes the input signal, $R(s)$, undiminished, to several output points.

We will now examine some common topologies for interconnecting subsystems and derive the single transfer function representation for each of them. These common topologies will form the basis for reducing more complicated systems to a single block.

Cascade Form

Figure 5.3(a) shows an example of cascaded subsystems. Intermediate signal values are shown at the output of each subsystem. Each signal is derived from the product of the input times the transfer function. The equivalent transfer function, $G_e(s)$, shown in Figure 5.3(b), is the output Laplace transform divided by the input Laplace

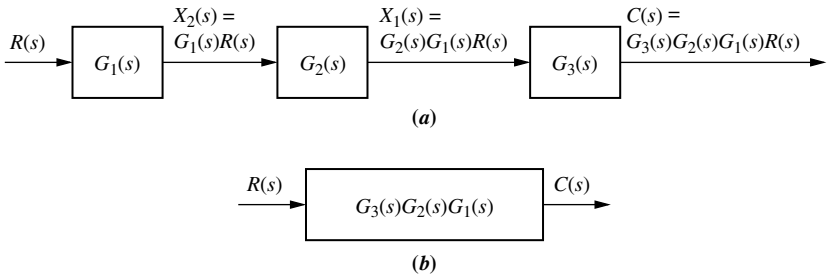


FIGURE 5.3 **a.** Cascaded subsystems; **b.** equivalent transfer function

transform from Figure 5.3(a), or

$$G_e(s) = G_3(s)G_2(s)G_1(s) \tag{5.1}$$

which is the product of the subsystems' transfer functions.

Equation (5.1) was derived under the assumption that interconnected subsystems do not load adjacent subsystems. That is, a subsystem's output remains the same whether or not the subsequent subsystem is connected. If there is a change in the output, the subsequent subsystem loads the previous subsystem, and the equivalent transfer function is not the product of the individual transfer functions. The network of Figure 5.4(a) demonstrates this concept. Its transfer function is

$$G_1(s) = \frac{V_1(s)}{V_i(s)} = \frac{\dfrac{1}{R_1C_1}}{s + \dfrac{1}{R_1C_1}} \tag{5.2}$$

Similarly, the network of Figure 5.4(b) has the following transfer function:

$$G_2(s) = \frac{V_2(s)}{V_1(s)} = \frac{\dfrac{1}{R_2C_2}}{s + \dfrac{1}{R_2C_2}} \tag{5.3}$$

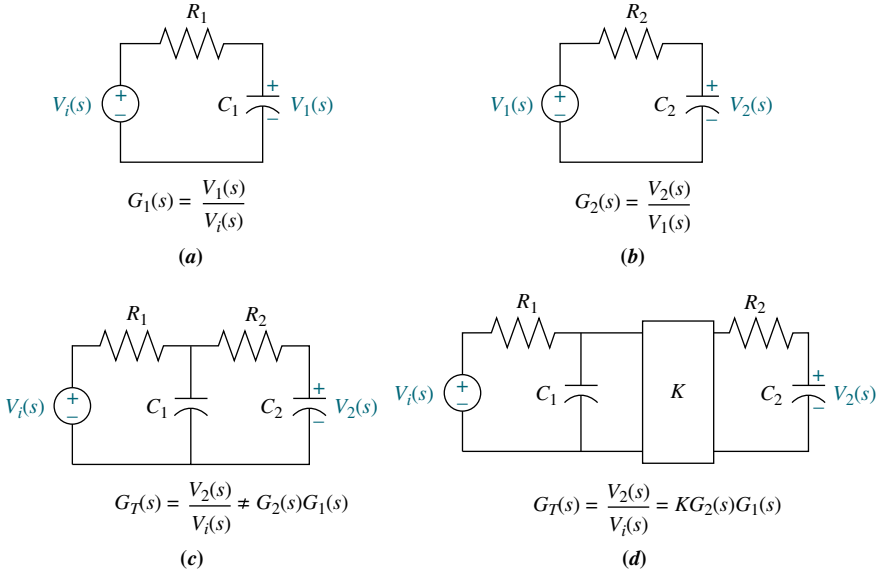


FIGURE 5.4 Loading in cascaded systems

If the networks are placed in cascade, as in Figure 5.4(c), you can verify that the transfer function found using loop or node equations is

$$G(s) = \frac{V_2(s)}{V_i(s)} = \frac{\dfrac{1}{R_1C_1R_2C_2}}{s^2 + \left(\dfrac{1}{R_1C_1} + \dfrac{1}{R_2C_2} + \dfrac{1}{R_2C_1}\right)s + \dfrac{1}{R_1C_1R_2C_2}} \tag{5.4}$$

But, using Eq. (5.1),

$$G(s) = G_2(s)G_1(s) = \frac{\dfrac{1}{R_1C_1R_2C_2}}{s^2 + \left(\dfrac{1}{R_1C_1} + \dfrac{1}{R_2C_2}\right)s + \dfrac{1}{R_1C_1R_2C_2}} \tag{5.5}$$

Equations (5.4) and (5.5) are not the same: Eq. (5.4) has one more term for the coefficient of s in the denominator and is correct.

One way to prevent loading is to use an amplifier between the two networks, as shown in Figure 5.4(d). The amplifier has a high-impedance input, so that it does not load the previous network. At the same time it has a low-impedance output, so that it looks like a pure voltage source to the subsequent network. With the amplifier included, the equivalent transfer function is the product of the transfer functions and the gain, K, of the amplifier.

Parallel Form

Figure 5.5 shows an example of parallel subsystems. Again, by writing the output of each subsystem, we can find the equivalent transfer function. Parallel subsystems have a common input and an output formed by the algebraic sum of the outputs from all of the subsystems. The equivalent transfer function, $G_e(s)$, is the output transform divided by the input transform from Figure 5.5(a), or

$$\boxed{G_e(s) = \pm G_1(s) \pm G_2(s) \pm G_3(s)} \tag{5.6}$$

which is the algebraic sum of the subsystems' transfer functions; it appears in Figure 5.5(b).

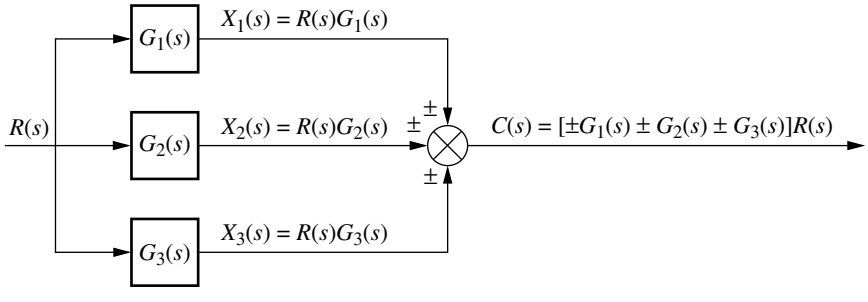


FIGURE 5.5 **a.** Parallel subsystems; **b.** equivalent transfer function

Feedback Form

The third topology is the feedback form, which will be seen repeatedly in subsequent chapters. The feedback system forms the basis for our study of control systems engineering. In Chapter 1, we defined open-loop and closed-loop systems and pointed out the advantage of closed-loop, or feedback control, systems over open-loop systems. As we move ahead, we will focus on the analysis and design of feedback systems.

Let us derive the transfer function that represents the system from its input to its output. The typical feedback system, described in detail in Chapter 1, is shown in Figure 5.6(a); a simplified model is shown in Figure 5.6(b).[1] Directing our attention to the simplified model,

$$E(s) = R(s) \mp C(s)H(s) \tag{5.7}$$

But since $C(s) = E(s)G(s)$,

$$E(s) = \frac{C(s)}{G(s)} \tag{5.8}$$

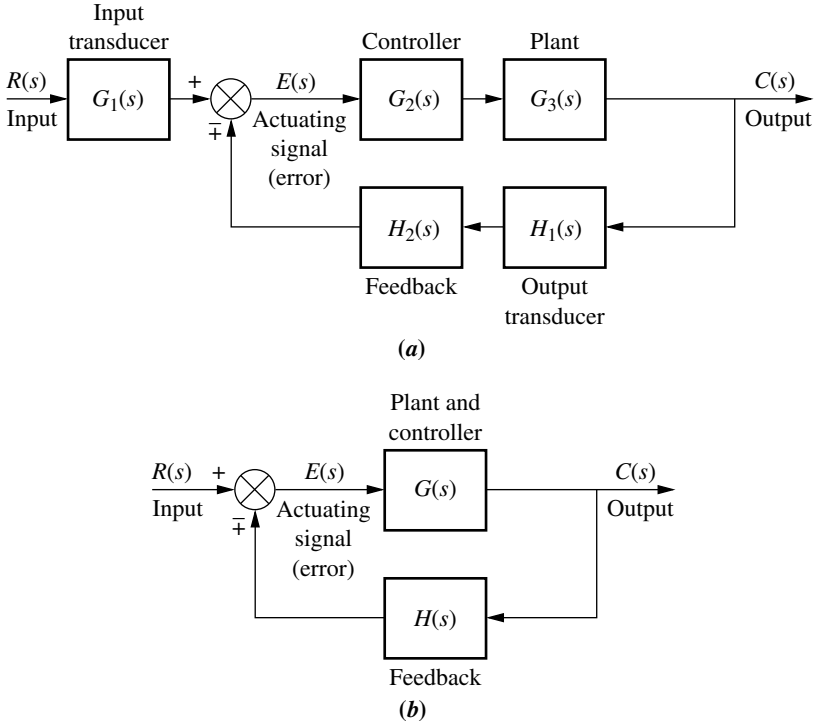


FIGURE 5.6 **a.** Feedback control system; **b.** simplified model; **c.** equivalent transfer function

[1] The system is said to have *negative feedback* if the sign at the summing junction is negative and *positive feedback* if the sign is positive.

Substituting Eq. (5.8) into Eq. (5.7) and solving for the transfer function, $C(s)/R(s) = G_e(s)$, we obtain the equivalent, or *closed-loop*, transfer function shown in Figure 5.6(*c*),

$$G_e(s) = \frac{G(s)}{1 \pm G(s)H(s)} \tag{5.9}$$

The product, $G(s)H(s)$, in Eq. (5.9) is called the *open-loop transfer function*, or *loop gain.*

So far, we have explored three different configurations for multiple subsystems. For each, we found the equivalent transfer function. Since these three forms are combined into complex arrangements in physical systems, recognizing these topologies is a prerequisite to obtaining the equivalent transfer function of a complex system. In this section, we will reduce complex systems composed of multiple subsystems to single transfer functions.

Moving Blocks to Create Familiar Forms

Before we begin to reduce block diagrams, it must be explained that the familiar forms (cascade, parallel, and feedback) are not always apparent in a block diagram. For example, in the feedback form, if there is a pickoff point after the summing junction, you cannot use the feedback formula to reduce the feedback system to a single block. That signal disappears, and there is no place to reestablish the pickoff point.

This subsection will discuss basic block moves that can be made to order to establish familiar forms when they almost exist. In particular, it will explain how to move blocks left and right past summing junctions and pickoff points.

Figure 5.7 shows equivalent block diagrams formed when transfer functions are moved left or right past a summing junction, and Figure 5.8 shows equivalent block diagrams formed when transfer functions are moved left or right past a pickoff point. In the diagrams the symbol ≡ means "equivalent to." These equivalences,

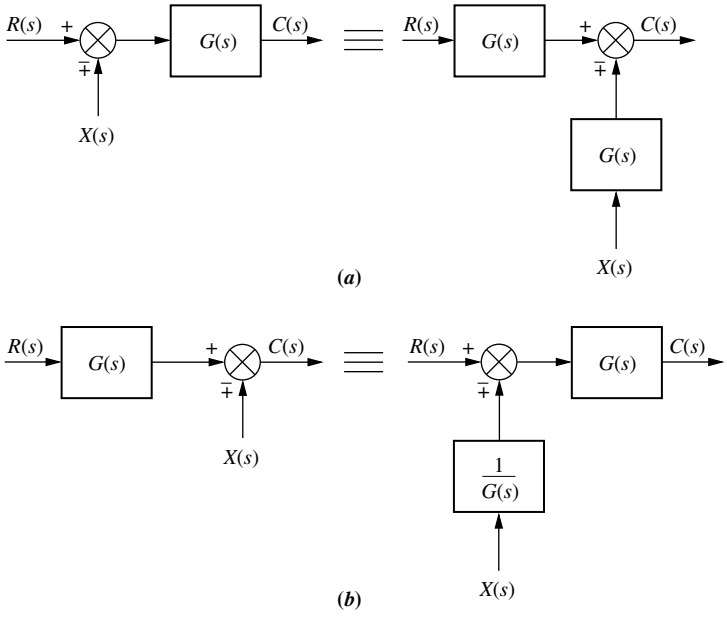


FIGURE 5.7 Block diagram algebra for summing junctions—equivalent forms for moving a block **a.** to the left past a summing junction; **b.** to the right past a summing junction

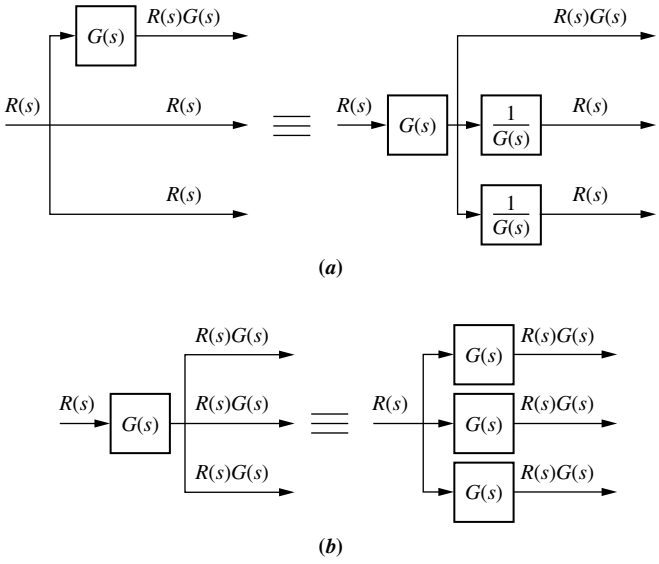


FIGURE 5.8 Block diagram algebra for pickoff points—equivalent forms for moving a block **a.** to the left past a pickoff point; **b.** to the right past a pickoff point

along with the forms studied earlier in this section, can be used to reduce a block diagram to a single transfer function. In each case of Figures 5.7 and 5.8, the equivalence can be verified by tracing the signals at the input through to the output and recognizing that the output signals are identical. For example, in Figure 5.7(a), signals $R(s)$ and $X(s)$ are multiplied by $G(s)$ before reaching the output. Hence, both block diagrams are equivalent, with $C(s) = R(s)G(s) \mp X(s)G(s)$. In Figure 5.7($b$), $R(s)$ is multiplied by $G(s)$ before reaching the output, but $X(s)$ is not. Hence, both block diagrams in Figure 5.7(b) are equivalent, with $C(s) = R(s)G(s) \mp X(s)$. For pickoff points, similar reasoning yields similar results for the block diagrams of Figure 5.8(a) and (b).

Let us now put the whole story together with examples of block diagram reduction.

Example 5.1

Block Diagram Reduction via Familiar Forms

PROBLEM: Reduce the block diagram shown in Figure 5.9 to a single transfer function.

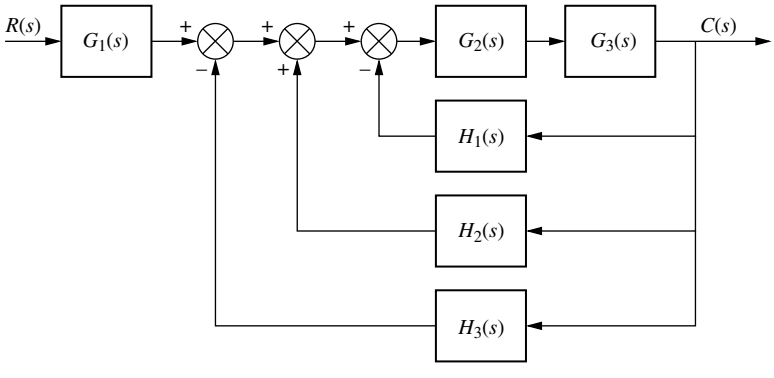


FIGURE 5.9 Block diagram for Example 5.1

SOLUTION: We solve the problem by following the steps in Figure 5.10. First, the three summing junctions can be collapsed into a single summing junction, as shown in Figure 5.10(*a*).

Second, recognize that the three feedback functions, $H_1(s)$, $H_2(s)$, and $H_3(s)$, are connected in parallel. They are fed from a common signal source, and their outputs are summed. The equivalent function is $H_1(s) - H_2(s) + H_3(s)$. Also recognize that $G_2(s)$ and $G_3(s)$ are connected in cascade. Thus, the equivalent transfer function is the product, $G_3(s)G_2(s)$. The results of these steps are shown in Figure 5.10(*b*).

Finally, the feedback system is reduced and multiplied by $G_1(s)$ to yield the equivalent transfer function shown in Figure 5.10(*c*).

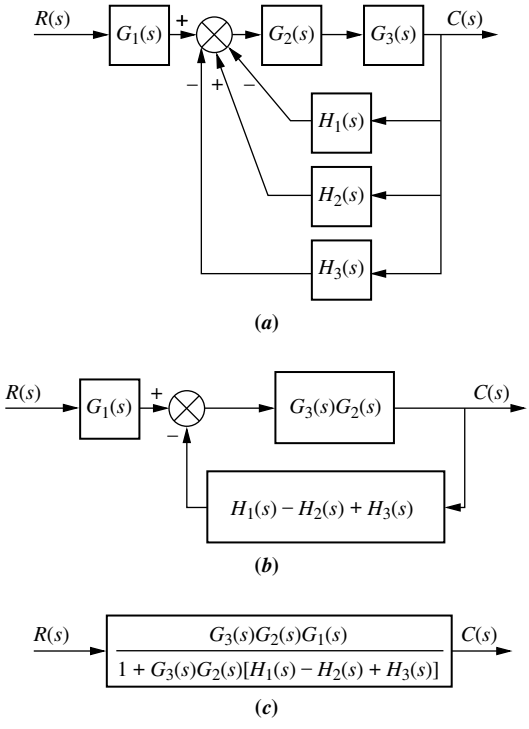


FIGURE 5.10 Steps in solving Example 5.1: **a.** Collapse summing junctions; **b.** form equivalent cascaded system in the forward path and equivalent parallel system in the feedback path; **c.** form equivalent feedback system and multiply by cascaded $G_1(s)$

Example 5.2

Block Diagram Reduction by Moving Blocks

PROBLEM: Reduce the system shown in Figure 5.11 to a single transfer function.

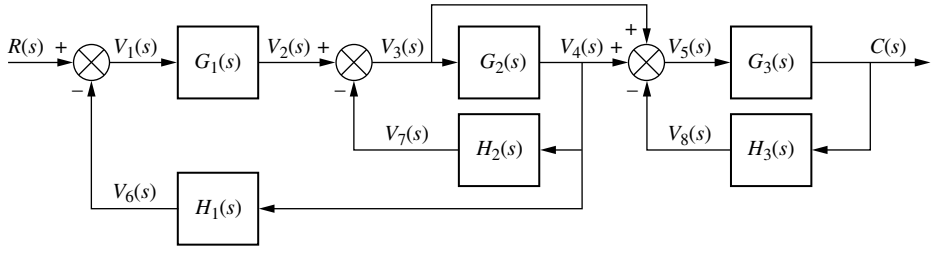
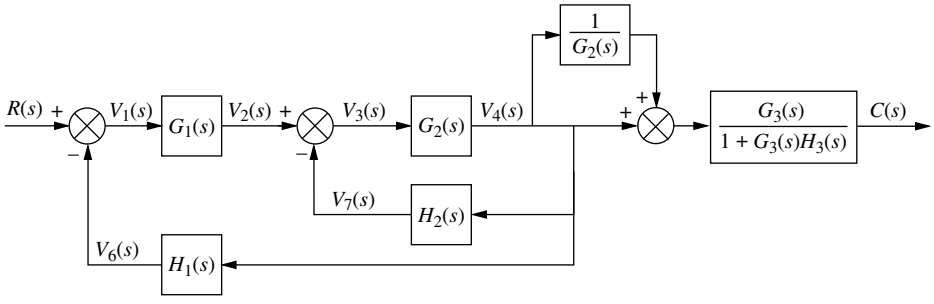


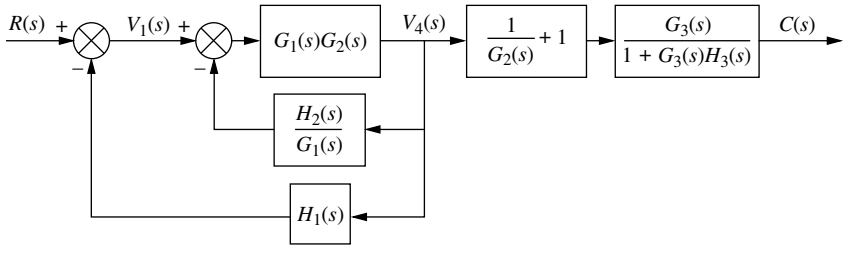
FIGURE 5.11 Block diagram for Example 5.2

SOLUTION: In this example we make use of the equivalent forms shown in Figures 5.7 and 5.8. First, move $G_2(s)$ to the left past the pickoff point to create parallel subsystems, and reduce the feedback system consisting of $G_3(s)$ and $H_3(s)$. This result is shown in Figure 5.12(a).

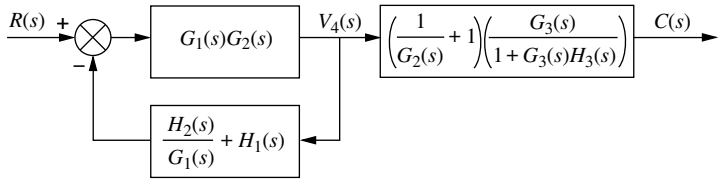
Second, reduce the parallel pair consisting of $1/G_2(s)$ and unity, and push $G_1(s)$ to the right past the summing junction, creating parallel subsystems in the feedback. These results are shown in Figure 5.12(b).



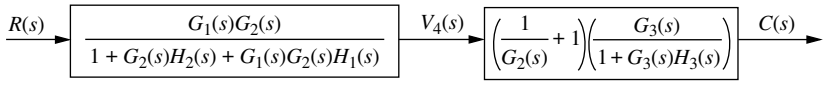
(a)



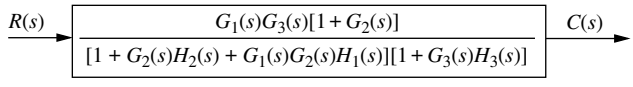
(b)



(c)



(d)



(e)

FIGURE 5.12 Steps in the block diagram reduction for Example 5.2

Third, collapse the summing junctions, add the two feedback elements together, and combine the last two cascaded blocks. Figure 5.12(*c*) shows these results.

Fourth, use the feedback formula to obtain Figure 5.12(*d*).

Finally, multiply the two cascaded blocks and obtain the final result, shown in Figure 5.12(*e*).

MATLAB

ML

Students who are using MATLAB should now run ch5p1 in Appendix B to perform block diagram reduction.

Skill-Assessment Exercise 5.1

PROBLEM: Find the equivalent transfer function, $T(s) = C(s)/R(s)$, for the system shown in Figure 5.13.

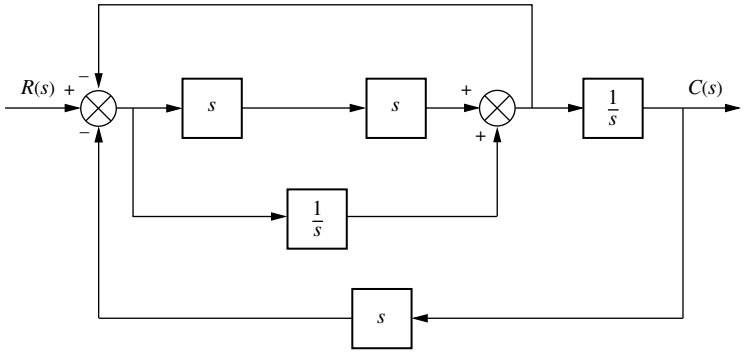


FIGURE 5.13 Block diagram for Skill-Assessment Exercise 5.1

ANSWER:

$$T(s) = \frac{s^3 + 1}{2s^4 + s^2 + 2s}$$

The complete solution is at www.wiley.com/college/nise.

TryIt 5.1

Use the following MATLAB and Control System Toolbox statements to find the closed-loop transfer function of the system in Example 5.2 if all $G_i(s) = 1/(s+1)$ and all $H_i(s) = 1/s$.

```
G1=tf(1,[1 1]);
G2=G1;G3=G1;
H1=tf(1,[1 0]);
H2=H1;H3=H1;
System=append...
 (G1,G2,G3,H1,H2,H3);
input=1;output=3;
Q=[1  -4   0   0   0
   2   1  -5   0   0
   3   2   1  -5  -6
   4   2   0   0   0
   5   2   0   0   0
   6   3   0   0   0];
T=connect(System,...
 Q,input,output);
T=tf(T);T=minreal(T)
```

In this section, we examined the equivalence of several block diagram configurations containing signals, systems, summing junctions, and pickoff points. These configurations were the cascade, parallel, and feedback forms. During block diagram reduction, we attempt to produce these easily recognized forms and then reduce the block diagram to a single transfer function. In the next section, we will examine some applications of block diagram reduction.

5.3 Analysis and Design of Feedback Systems

An immediate application of the principles of Section 5.2 is the analysis and design of feedback systems that reduce to second-order systems. Percent overshoot, settling time, peak time, and rise time can then be found from the equivalent transfer function.

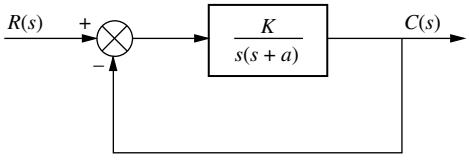


FIGURE 5.14 Second-order feedback control system

Consider the system shown in Figure 5.14, which can model a control system such as the antenna azimuth position control system. For example, the transfer function, $K/s(s+a)$, can model the amplifiers, motor, load, and gears. From Eq. (5.9), the closed-loop transfer function, $T(s)$, for this system is

$$T(s) = \frac{K}{s^2 + as + K} \tag{5.10}$$

where K models the amplifier gain, that is, the ratio of the output voltage to the input voltage. As K varies, the poles move through the three ranges of operation of a second-order system: overdamped, critically damped, and underdamped. For example, for K between 0 and $a^2/4$, the poles of the system are real and are located at

$$s_{1,2} = -\frac{a}{2} \pm \frac{\sqrt{a^2 - 4K}}{2} \tag{5.11}$$

As K increases, the poles move along the real axis, and the system remains overdamped until $K = a^2/4$. At that gain, or amplification, both poles are real and equal, and the system is critically damped.

For gains above $a^2/4$, the system is underdamped, with complex poles located at

$$s_{1,2} = -\frac{a}{2} \pm j\frac{\sqrt{4K - a^2}}{2} \tag{5.12}$$

Now as K increases, the real part remains constant and the imaginary part increases. Thus, the peak time decreases and the percent overshoot increases, while the settling time remains constant.

Let us look at two examples that apply the concepts to feedback control systems. In the first example, we determine a system's transient response. In the second example, we design the gain to meet a transient response requirement.

Example 5.3

Finding Transient Response

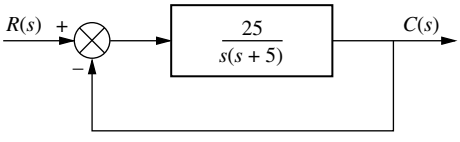


FIGURE 5.15 Feedback system for Example 5.3

PROBLEM: For the system shown in Figure 5.15, find the peak time, percent overshoot, and settling time.

SOLUTION: The closed-loop transfer function found from Eq. (5.9) is

$$T(s) = \frac{25}{s^2 + 5s + 25} \tag{5.13}$$

From Eq. (4.18),

$$\omega_n = \sqrt{25} = 5 \tag{5.14}$$

From Eq. (4.21),

$$2\zeta\omega_n = 5 \tag{5.15}$$

Substituting Eq. (5.14) into (5.15) and solving for ζ yields

$$\zeta = 0.5 \tag{5.16}$$

Using the values for ζ and ω_n along with Eqs (4.34), (4.38), and (4.42), we find respectively,

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = 0.726 \text{ second} \tag{5.17}$$

$$\%OS = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \times 100 = 16.303 \tag{5.18}$$

$$T_s = \frac{4}{\zeta\omega_n} = 1.6 \text{ seconds} \tag{5.19}$$

MATLAB
ML

Students who are using MATLAB should now run ch5p2 in Appendix B. You will learn how to perform block diagram reduction followed by an evaluation of the closed-loop system's transient response by finding, T_p, $\%OS$, and T_s. Finally, you will learn how to use MATLAB to generate a closed-loop step response. This exercise uses MATLAB to do Example 5.3.

Simulink
SL

MATLAB's Simulink provides an alternative method of simulating feedback systems to obtain the time response. Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Simulink should now consult Appendix C. Example C.3 includes a discussion about, and an example of, the use of Simulink to simulate feedback systems with nonlinearities.

Example 5.4

Gain Design for Transient Response

PROBLEM: Design the value of gain, K, for the feedback control system of Figure 5.16 so that the system will respond with a 10% overshoot.

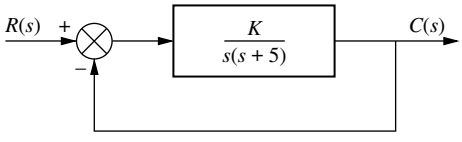


FIGURE 5.16 Feedback system for Example 5.4

SOLUTION: The closed-loop transfer function of the system is

$$T(s) = \frac{K}{s^2 + 5s + K} \tag{5.20}$$

From Eq. (5.20),

$$2\zeta\omega_n = 5 \tag{5.21}$$

and

$$\omega_n = \sqrt{K} \tag{5.22}$$

Thus,

$$\zeta = \frac{5}{2\sqrt{K}} \tag{5.23}$$

Since percent overshoot is a function only of ζ, Eq. (5.23) shows that the percent overshoot is a function of K.

Virtual Experiment 5.1
Position Control Gain Design

Put theory into practice designing the position control gain for the Quanser Linear Servo and simulating its closed-loop response in LabVIEW. This concept is used, for instance, to control a rover exploring the terrain of a planet.



Virtual experiment are found on WileyPLUS.

A 10% overshoot implies that $\zeta = 0.591$. Substituting this value for the damping ratio into Eq. (5.23) and solving for K yields

$$K = 17.9 \tag{5.24}$$

Although we are able to design for percent overshoot in this problem, we could not have selected settling time as a design criterion because, regardless of the value of K, the real parts, -2.5, of the poles of Eq. (5.20) remain the same.

Skill-Assessment Exercise 5.2

WileyPLUS
WPCS
Control Solutions

PROBLEM: For a unity feedback control system with a forward-path transfer function $G(s) = \dfrac{16}{s(s+a)}$, design the value of a to yield a closed-loop step response that has 5% overshoot.

ANSWER:

$$a = 5.52$$

The complete solution is at www.wiley.com/college/nise.

TryIt 5.2

Use the following MATLAB and Control System Toolbox statements to find ζ, ω_n, $\%OS$, T_s, T_p, and T_r for the closed-loop unity feedback system described in Skill-Assessment Exercise 5.2. Start with $a = 2$ and try some other values. A step response for the closed-loop system will also be produced.

```
a=2;
numg=16;
deng=poly([0 -a]);
G=tf(numg,deng);
T=feedback(G,1);
```

```
[numt,dent]=...
 tfdata(T,'v');
wn=sqrt(dent(3))
z=dent(2)/(2*wn)
Ts=4/(z*wn)
Tp=pi/(wn*...
 sqrt(1-z^2))
pos=exp(-z*pi...
 /sqrt(1-z^2))*100
Tr=(1.76*z^3-...
 0.417*z^2+1.039*...
 z+1)/wn
step(T)
```

5.4 Signal-Flow Graphs

Signal-flow graphs are an alternative to block diagrams. Unlike block diagrams, which consist of blocks, signals, summing junctions, and pickoff points, a signal-flow graph consists only of *branches*, which represent systems, and *nodes*, which represent signals. These elements are shown in Figure 5.17(*a*) and (*b*), respectively. A system is represented by a line with an arrow showing the direction of signal flow through the

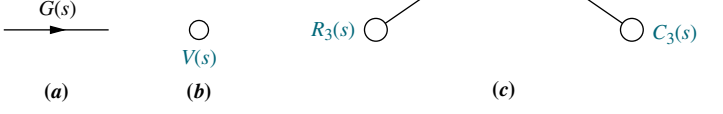


FIGURE 5.17 Signal-flow graph components: **a.** system; **b.** signal; **c.** interconnection of systems and signals

system. Adjacent to the line we write the transfer function. A signal is a node with the signal's name written adjacent to the node.

Figure 5.17(*c*) shows the interconnection of the systems and the signals. Each signal is the sum of signals flowing into it. For example, the signal $V(s) = R_1(s)G_1(s) - R_2(s)G_2(s) + R_3(s)G_3(s)$. The signal $C_2(s) = V(s)G_5(s) = R_1(s)G_1(s)G_5(s) - R_2(s)G_2(s)G_5(s) + R_3(s)G_3(s)G_5(s)$. The signal $C_3(s) = -V(s)G_6(s) = -R_1(s)G_1(s)G_6(s) + R_2(s)G_2(s)G_6(s) - R_3(s)G_3(s)G_6(s)$. Notice that in summing negative signals we associate the negative sign with the system and not with a summing junction, as in the case of block diagrams.

To show the parallel between block diagrams and signal-flow graphs, we will take some of the block diagram forms from Section 5.2 and convert them to signal-flow graphs in Example 5.5. In each case, we will first convert the signals to nodes and then interconnect the nodes with system branches. In Example 5.6, we will convert an intricate block diagram to a signal-flow graph.

Example 5.5

Converting Common Block Diagrams to Signal-Flow Graphs

PROBLEM: Convert the cascaded, parallel, and feedback forms of the block diagrams shown in Figures 5.3(*a*), 5.5(*a*), and 5.6(*b*), respectively, into signal-flow graphs.

SOLUTION: In each case, we start by drawing the signal nodes for that system. Next we interconnect the signal nodes with system branches. The signal nodes for the cascaded, parallel, and feedback forms are shown in Figure 5.18(*a*), (*c*), and (*e*), respectively. The interconnection of the nodes with branches that represent the subsystems is shown in Figure 5.18(*b*), (*d*), and (*f*) for the cascaded, parallel, and feedback forms, respectively.

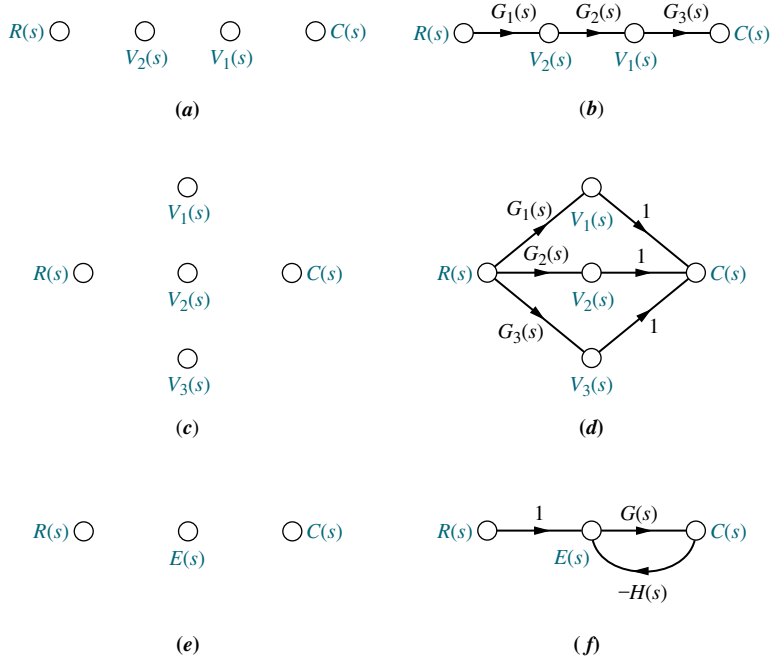


FIGURE 5.18 Building signal-flow graphs: **a.** cascaded system nodes (from Figure 5.3(*a*)); **b.** cascaded system signal-flow graph; **c.** parallel system nodes (from Figure 5.5(*a*)); **d.** parallel system signal-flow graph; **e.** feedback system nodes (from Figure 5.6(*b*)); **f.** feedback system signal-flow graph

Example 5.6

Converting a Block Diagram to a Signal-Flow Graph

PROBLEM: Convert the block diagram of Figure 5.11 to a signal-flow graph.

SOLUTION: Begin by drawing the signal nodes, as shown in Figure 5.19(*a*). Next, interconnect the nodes, showing the direction of signal flow and identifying each transfer function. The result is shown in Figure 5.19(*b*). Notice that the negative signs at the summing junctions of the block diagram are represented by the negative transfer functions of the signal-flow graph. Finally, if desired, simplify the signal-flow graph to the one shown in Figure 5.19(*c*) by eliminating signals that have a single flow in and a single flow out, such as $V_2(s)$, $V_6(s)$, $V_7(s)$, and $V_8(s)$.

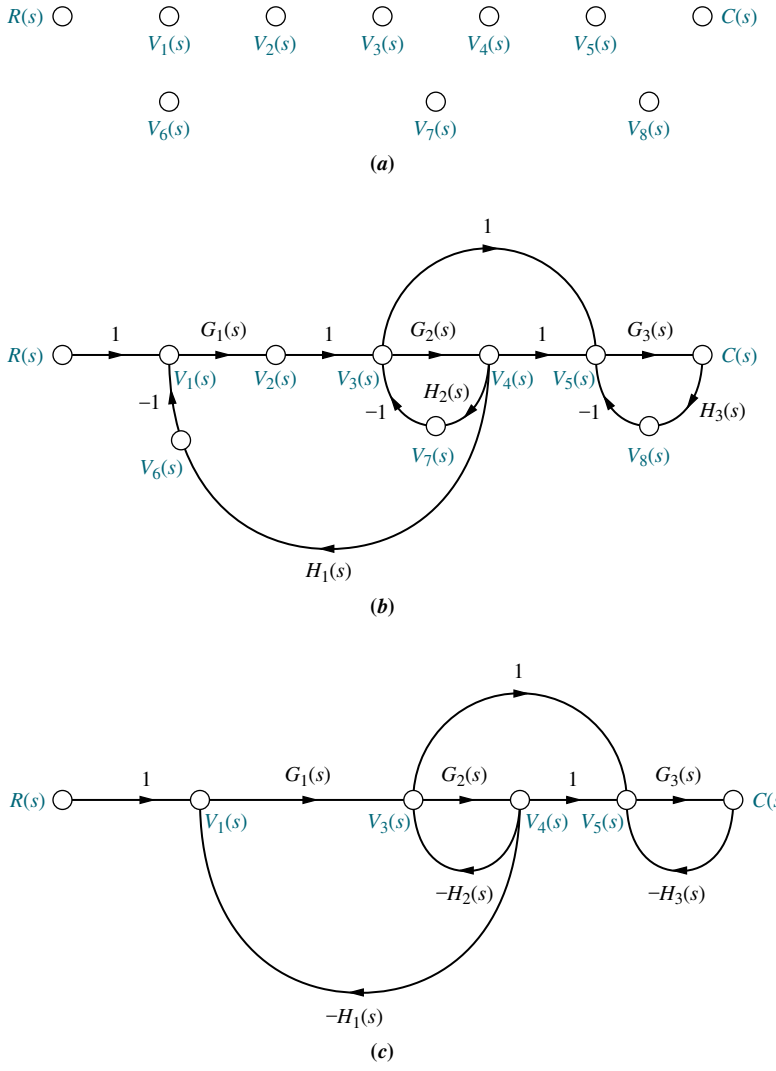


FIGURE 5.19 Signal-flow graph development: **a.** signal nodes; **b.** signal-flow graph; **c.** simplified signal-flow graph

Skill-Assessment Exercise 5.3

PROBLEM: Convert the block diagram of Figure 5.13 to a signal-flow graph.

ANSWER: The complete solution is at www.wiley.com/college/nise.

5.5 Mason's Rule

Earlier in this chapter, we discussed how to reduce block diagrams to single transfer functions. Now we are ready to discuss a technique for reducing signal-flow graphs to single transfer functions that relate the output of a system to its input.

The block diagram reduction technique we studied in Section 5.2 requires successive application of fundamental relationships in order to arrive at the system transfer function. On the other hand, Mason's rule for reducing a signal-flow graph to a single transfer function requires the application of one formula. The formula was derived by S. J. Mason when he related the signal-flow graph to the simultaneous equations that can be written from the graph (*Mason*, *1953*).

In general, it can be complicated to implement the formula without making mistakes. Specifically, the existence of what we will later call nontouching loops increases the complexity of the formula. However, many systems do not have nontouching loops. For these systems, you may find Mason's rule easier to use than block diagram reduction.

Mason's formula has several components that must be evaluated. First, we must be sure that the definitions of the components are well understood. Then we must exert care in evaluating the components. To that end, we discuss some basic definitions applicable to signal-flow graphs; then we state Mason's rule and do an example.

Definitions

Loop gain. The product of branch gains found by traversing a path that starts at a node and ends at the same node, following the direction of the signal flow, without passing through any other node more than once. For examples of loop gains, see Figure 5.20. There are four loop gains:

1. $G_2(s)H_1(s)$ (5.25a)
2. $G_4(s)H_2(s)$ (5.25b)
3. $G_4(s)G_5(s)H_3(s)$ (5.25c)
4. $G_4(s)G_6(s)H_3(s)$ (5.25d)

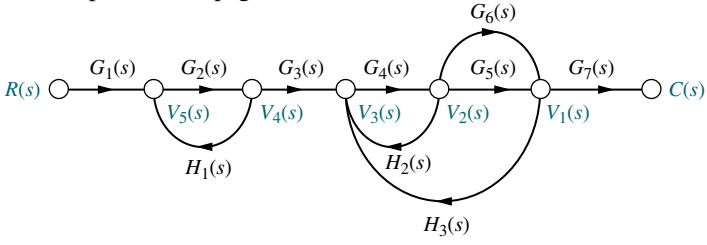


FIGURE 5.20 Signal-flow graph for demonstrating Mason's rule

Forward-path gain. The product of gains found by traversing a path from the input node to the output node of the signal-flow graph in the direction of signal flow. Examples of forward-path gains are also shown in Figure 5.20. There are two forward-path gains:

1. $G_1(s)G_2(s)G_3(s)G_4(s)G_5(s)G_7(s)$ (5.26a)
2. $G_1(s)G_2(s)G_3(s)G_4(s)G_6(s)G_7(s)$ (5.26b)

Nontouching loops. Loops that do not have any nodes in common. In Figure 5.20, loop $G_2(s)H_1(s)$ does not touch loops $G_4(s)H_2(s)$, $G_4(s)G_5(s)H_3(s)$, and $G_4(s)G_6(s)H_3(s)$.

Nontouching-loop gain. The product of loop gains from nontouching loops taken two, three, four, or more at a time. In Figure 5.20 the product of loop gain $G_2(s)H_1(s)$ and loop gain $G_4(s)H_2(s)$ is a nontouching-loop gain taken two at a time. In summary, all three of the nontouching-loop gains taken two at a time are

1. $[G_2(s)H_1(s)][G_4(s)H_2(s)]$ (5.27a)

2. $[G_2(s)H_1(s)][G_4(s)G_5(s)H_3(s)]$ (5.27b)

3. $[G_2(s)H_1(s)][G_4(s)G_6(s)H_3(s)]$ (5.27c)

The product of loop gains $[G_4(s)G_5(s)H_3(s)][G_4(s)G_6(s)H_3(s)]$ is not a nontouching-loop gain since these two loops have nodes in common. In our example there are no nontouching-loop gains taken three at a time since three nontouching loops do not exist in the example.

We are now ready to state Mason's rule.

Mason's Rule

The transfer function, $C(s)/R(s)$, of a system represented by a signal-flow graph is

$$G(s) = \frac{C(s)}{R(s)} = \frac{\sum_k T_k \Delta_k}{\Delta} \tag{5.28}$$

where

k = number of forward paths
T_k = the kth forward-path gain
Δ = $1 - \Sigma$ loop gains $+ \Sigma$ nontouching-loop gains taken two at a time $- \Sigma$ nontouching-loop gains taken three at a time $+ \Sigma$ nontouching-loop gains taken four at a time $- \ldots$
Δ_k = $\Delta - \Sigma$ loop gain terms in Δ that touch the kth forward path. In other words, Δ_k is formed by eliminating from Δ those loop gains that touch the kth forward path.

Notice the alternating signs for the components of Δ. The following example will help clarify Mason's rule.

Example 5.7

Transfer Function via Mason's Rule

PROBLEM: Find the transfer function, $C(s)/R(s)$, for the signal-flow graph in Figure 5.21.

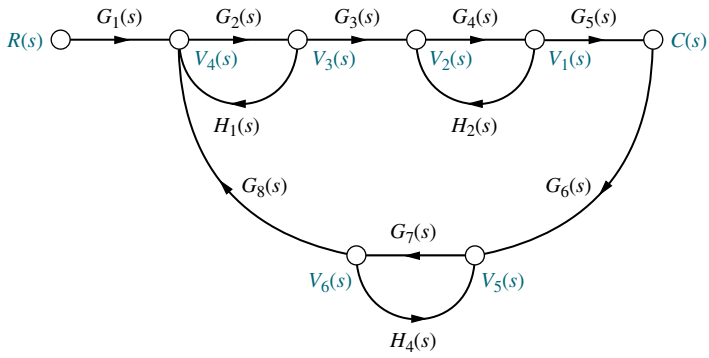


FIGURE 5.21 Signal-flow graph for Example 5.7

SOLUTION: First, identify the *forward-path gains*. In this example there is only one:

$$G_1(s)G_2(s)G_3(s)G_4(s)G_5(s) \tag{5.29}$$

Second, identify the *loop gains*. There are four, as follows:

1. $G_2(s)H_1(s)$ (5.30a)

2. $G_4(s)H_2(s)$ (5.30b)

3. $G_7(s)H_4(s)$ (5.30c)

4. $G_2(s)G_3(s)G_4(s)G_5(s)G_6(s)G_7(s)G_8(s)$ (5.30d)

Third, identify the *nontouching loops taken two at a time*. From Eqs. (5.30) and Figure 5.21, we can see that loop 1 does not touch loop 2, loop 1 does not touch loop 3, and loop 2 does not touch loop 3. Notice that loops 1, 2, and 3 all touch loop 4. Thus, the combinations of nontouching loops taken two at a time are as follows:

$$\text{Loop 1 and loop 2}: \; G_2(s)H_1(s)G_4(s)H_2(s) \tag{5.31a}$$

$$\text{Loop 1 and loop 3}: \; G_2(s)H_1(s)G_7(s)H_4(s) \tag{5.31b}$$

$$\text{Loop 2 and loop 3}: \; G_4(s)H_2(s)G_7(s)H_4(s) \tag{5.31c}$$

Finally, the *nontouching loops taken three at a time* are as follows:

$$\text{Loops 1, 2, and 3}: \; G_2(s)H_1(s)G_4(s)H_2(s)G_7(s)H_4(s) \tag{5.32}$$

Now, from Eq. (5.28) and its definitions, we form Δ and Δ_k. Hence,

$$\begin{aligned}\Delta = 1 \; &-[G_2(s)H_1(s) + G_4(s)H_2(s) + G_7(s)H_4(s) \\ &\quad + G_2(s)G_3(s)G_4(s)G_5(s)G_6(s)G_7(s)G_8(s)] \\ &+[G_2(s)H_1(s)G_4(s)H_2(s) + G_2(s)H_1(s)G_7(s)H_4(s) \\ &\quad + G_4(s)H_2(s)G_7(s)H_4(s)] \\ &-[G_2(s)H_1(s)G_4(s)H_2(s)G_7(s)H_4(s)]\end{aligned} \tag{5.33}$$

We form Δ_k by eliminating from Δ the loop gains that touch the kth forward path:

$$\Delta_1 = 1 - G_7(s)H_4(s) \tag{5.34}$$

Expressions (5.29), (5.33), and (5.34) are now substituted into Eq. (5.28), yielding the transfer function:

$$G(s) = \frac{T_1\Delta_1}{\Delta} = \frac{[G_1(s)G_2(s)G_3(s)G_4(s)G_5(s)][1 - G_7(s)H_4(s)]}{\Delta} \tag{5.35}$$

Since there is only one forward path, $G(s)$ consists of only one term, rather than a sum of terms, each coming from a forward path.

Skill-Assessment Exercise 5.4

WileyPLUS
WPCS
Control Solutions

PROBLEM: Use Mason's rule to find the transfer function of the signal-flow diagram shown in Figure 5.19(*c*). Notice that this is the same system used in Example 5.2 to find the transfer function via block diagram reduction.

ANSWER:

$$T(s) = \frac{G_1(s)G_3(s)[1 + G_2(s)]}{[1 + G_2(s)H_2(s) + G_1(s)G_2(s)H_1(s)][1 + G_3(s)H_3(s)]}$$

The complete solution is at www.wiley.com/college/nise.

5.6 Signal-Flow Graphs of State Equations

State Space
SS

In this section, we draw signal-flow graphs from state equations. At first this process will help us visualize state variables. Later we will draw signal-flow graphs and then write alternate representations of a system in state space.

Consider the following state and output equations:

$$\dot{x}_1 = 2x_1 - 5x_2 + 3x_3 + 2r \tag{5.36a}$$

$$\dot{x}_2 = -6x_1 - 2x_2 + 2x_3 + 5r \tag{5.36b}$$

$$\dot{x}_3 = x_1 - 3x_2 - 4x_3 + 7r \tag{5.36c}$$

$$y = -4x_1 + 6x_2 + 9x_3 \tag{5.36d}$$

First, identify three nodes to be the three state variables, x_1, x_2, and x_3; also identify three nodes, placed to the left of each respective state variable, to be the derivatives of the state variables, as in Figure 5.22(*a*). Also identify a node as the input, r, and another node as the output, y.

Next interconnect the state variables and their derivatives with the defining integration, $1/s$, as shown in Figure 5.22(*b*). Then using Eqs. (5.36), feed to each node the indicated signals. For example, from Eq. (5.36*a*), $\dot{x}_1$ receives $2x_1 - 5x_2 + 3x_3 + 2r$, as shown in Figure 5.22(*c*). Similarly, $\dot{x}_2$ receives $-6x_1 - 2x_2 + 2x_3 + 5r$, as shown in Figure 5.22(*d*), and $\dot{x}_3$ receives $x_1 - 3x_2 - 4x_3 + 7r$, as shown in Figure 5.22(*e*). Finally, using Eq. (5.36*d*), the output, y, receives $-4x_1 + 6x_2 + 9x_3$, as shown in Figure 5.19(*f*), the final phase-variable representation, where the state variables are the outputs of the integrators.

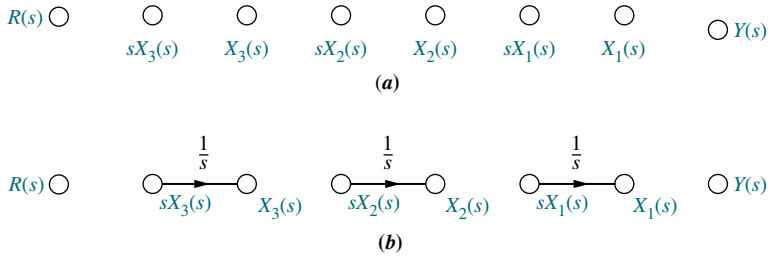
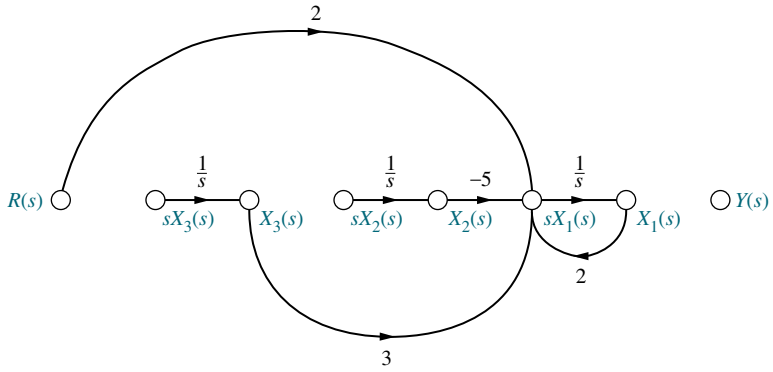
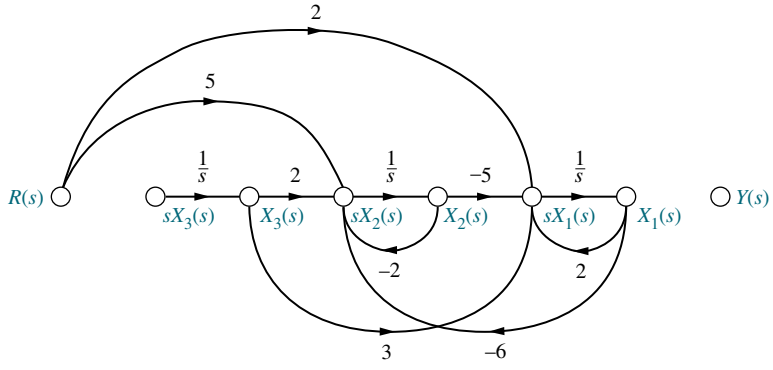


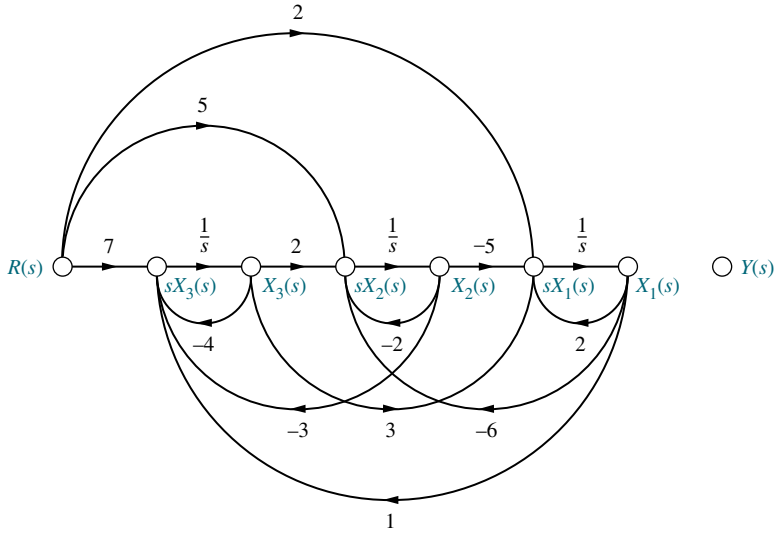
FIGURE 5.22 Stages of development of a signal-flow graph for the system of Eqs. (5.36): **a.** Place nodes; **b.** interconnect state variables and derivatives; (*figure continues*)



(*c*)

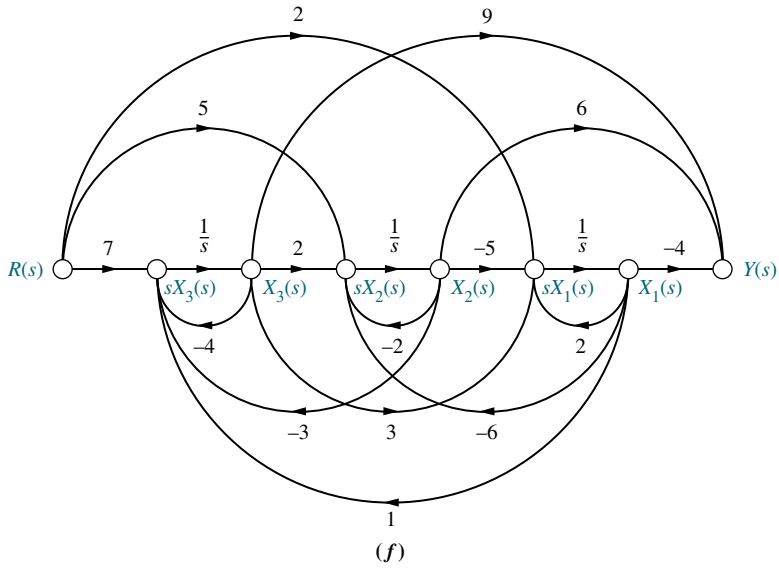


(*d*)



(*e*)

FIGURE 5.22 (*Continued*) **c.** form dx_1/dt; **d.** form dx_2/dt; **e.** form dx_3/dt; (*figure continues*)



(*f*)

FIGURE 5.22 (*Continued*) **f.** form output (*figure end*)

Skill-Assessment Exercise 5.5

PROBLEM: Draw a signal-flow graph for the following state and output equations:

$$\dot{\mathbf{x}} = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -3 & 1 \\ -3 & -4 & -5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

$$y = [0 \quad 1 \quad 0]\mathbf{x}$$

ANSWER: The complete solution is at www.wiley.com/college/nise.

In the next section, the signal-flow model will help us visualize the process of determining alternative representations in state space of the same system. We will see that even though a system can be the same with respect to its input and output terminals, the state-space representations can be many and varied.

5.7 Alternative Representations in State Space

State Space SS

In Chapter 3, systems were represented in state space in phase-variable form. However, system modeling in state space can take on many representations other than the phase-variable form. Although each of these models yields the same output for a given input, an engineer may prefer a particular one for several reasons. For example, one set of state variables, with its unique representation, can model actual physical variables of a system, such as amplifier and filter outputs.

Another motive for choosing a particular set of state variables and state-space model is ease of solution. As we will see, a particular choice of state variables can decouple the system of simultaneous differential equations. Here each equation is written in terms of only one state variable, and the solution is effected by solving n first-order differential equations individually.

Ease of modeling is another reason for a particular choice of state variables. Certain choices may facilitate converting the subsystem to the state-variable representation by using recognizable features of the model. The engineer learns quickly how to write the state and output equations and draw the signal-flow graph, both by inspection. These converted subsystems generate the definition of the state variables.

We will now look at a few representative forms and show how to generate the state-space representation for each.

Cascade Form

We have seen that systems can be represented in state space with the state variables chosen to be the phase variables, that is, variables that are successive derivatives of each other. This is by no means the only choice. Returning to the system of Figure 3.10(a), the transfer function can be represented alternately as

$$\frac{C(s)}{R(s)} = \frac{24}{(s+2)(s+3)(s+4)} \tag{5.37}$$

Figure 5.23 shows a block diagram representation of this system formed by cascading each term of Eq. (5.37). The output of each first-order system block has been labeled as a state variable. These state variables are not the phase variables.

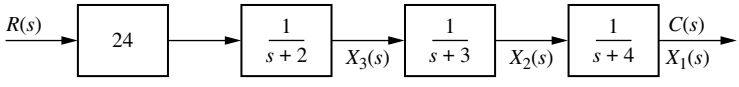


FIGURE 5.23 Representation of Figure 3.10 system as cascaded first-order systems

We now show how the signal-flow graph can be used to obtain a state-space representation of this system. In order to write the state equations with our new set of state variables, it is helpful to draw a signal-flow graph first, using Figure 5.23 as a guide. The signal flow for each first-order system of Figure 5.23 can be found by transforming each block into an equivalent differential equation. Each first-order block is of the form

$$\frac{C_i(s)}{R_i(s)} = \frac{1}{(s+a_i)} \tag{5.38}$$

Cross-multiplying, we get

$$(s+a_i)C_i(s) = R_i(s) \tag{5.39}$$

After taking the inverse Laplace transform, we have

$$\frac{dc_i(t)}{dt} + a_i c_i(t) = r_i(t) \tag{5.40}$$

Solving for $dc_i(t)/dt$ yields

$$\frac{dc_i(t)}{dt} = -a_i c_i(t) + r_i(t) \tag{5.41}$$

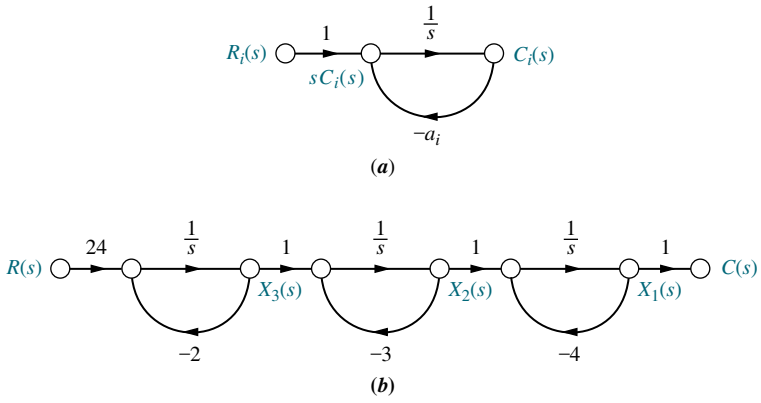


FIGURE 5.24 **a.** First-order subsystem; **b.** Signal-flow graph for Figure 5.23 system

Figure 5.24(a) shows the implementation of Eq. (5.41) as a signal-flow graph. Here again, a node was assumed for $c_i(t)$ at the output of an integrator, and its derivative was formed at the input.

Cascading the transfer functions shown in Figure 5.24(a), we arrive at the system representation shown in Figure 5.24(b).[2] Now write the state equations for the new representation of the system. Remember that the derivative of a state variable will be at the input to each integrator:

$$\dot{x}_1 = -4x_1 + x_2 \tag{5.42a}$$

$$\dot{x}_2 = -3x_2 + x_3 \tag{5.42b}$$

$$\dot{x}_3 = -2x_3 + 24r \tag{5.42c}$$

The output equation is written by inspection from Figure 5.24(b):

$$y = c(t) = x_1 \tag{5.43}$$

The state-space representation is completed by rewriting Eqs. (5.42) and (5.43) in vector-matrix form:

$$\dot{\mathbf{x}} = \begin{bmatrix} -4 & 1 & 0 \\ 0 & -3 & 1 \\ 0 & 0 & -2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 24 \end{bmatrix} r \tag{5.44a}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x} \tag{5.44b}$$

Comparing Eqs. (5.44) with Figure 5.24(b), you can form a vivid picture of the meaning of some of the components of the state equation. For the following discussion, please refer back to the general form of the state and output equations, Eqs. (3.18) and (3.19).

For example, the **B** matrix is the input matrix since it contains the terms that couple the input, $r(t)$, to the system. In particular, the constant 24 appears in both the signal-flow graph at the input, as shown in Figure 5.24(b), and the input matrix in Eqs. (5.44). The **C** matrix is the output matrix since it contains the constant that couples the state variable, x_1, to the output, $c(t)$. Finally, the **A** matrix is the system

[2] Note that node $X_3(s)$ and the following node cannot be merged, or else the input to the first integrator would be changed by the feedback from $X_2(s)$, and the signal $X_3(s)$ would be lost. A similar argument can be made for $X_2(s)$ and the following node.

matrix since it contains the terms relative to the internal system itself. In the form of Eqs. (5.44), the system matrix actually contains the system poles along the diagonal.

Compare Eqs. (5.44) to the phase-variable representation in Eqs. (3.59). In that representation, the coefficients of the system's characteristic polynomial appeared along the last row, whereas in our current representation, the roots of the characteristic equation, the system poles, appear along the diagonal.

Parallel Form

Another form that can be used to represent a system is the parallel form. This form leads to an **A** matrix that is purely diagonal, provided that no system pole is a repeated root of the characteristic equation.

Whereas the previous form was arrived at by cascading the individual first-order subsystems, the parallel form is derived from a partial-fraction expansion of the system transfer function. Performing a partial-fraction expansion on our example system, we get

$$\frac{C(s)}{R(s)} = \frac{24}{(s+2)(s+3)(s+4)} = \frac{12}{(s+2)} - \frac{24}{(s+3)} + \frac{12}{(s+4)} \tag{5.45}$$

Equation (5.45) represents the sum of the individual first-order subsystems. To arrive at a signal-flow graph, first solve for $C(s)$,

$$C(s) = R(s)\frac{12}{(s+2)} - R(s)\frac{24}{(s+3)} + R(s)\frac{12}{(s+4)} \tag{5.46}$$

and recognize that $C(s)$ is the sum of three terms. Each term is a first-order subsystem with $R(s)$ as the input. Formulating this idea as a signal-flow graph renders the representation shown in Figure 5.25.

Once again, we use the signal-flow graph as an aid to obtaining the state equations. By inspection the state variables are the outputs of each integrator, where the derivatives of the state variables exist at the integrator inputs. We write the state equations by summing the signals at the integrator inputs:

$$\dot{x}_1 = -2x_1 + 12r \tag{5.47a}$$

$$\dot{x}_2 = -3x_2 - 24r \tag{5.47b}$$

$$\dot{x}_3 = -4x_3 + 12r \tag{5.47c}$$

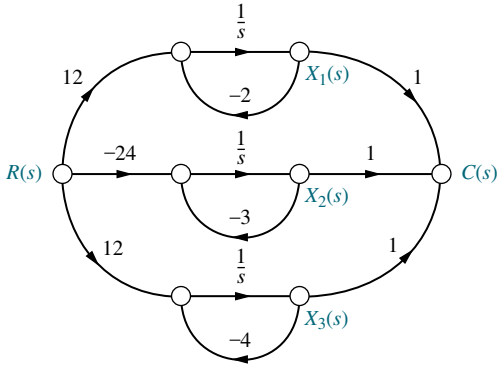


FIGURE 5.25 Signal-flow representation of Eq. (5.45)

The output equation is found by summing the signals that give $c(t)$:

$$y = c(t) = x_1 + x_2 + x_3 \tag{5.48}$$

In vector-matrix form, Eqs. (5.47) and (5.48) become

$$\dot{\mathbf{x}} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -4 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 12 \\ -24 \\ 12 \end{bmatrix} r \tag{5.49}$$

and

$$y = [1 \quad 1 \quad 1]\mathbf{x} \tag{5.50}$$

Thus, our third representation of the system of Figure 3.10(a) yields a diagonal system matrix. What is the advantage of this representation? Each equation is a

first-order differential equation in only one variable. Thus, we would solve these equations independently. The equations are said to be *decoupled.*

MATLAB
ML

Students who are using MATLAB should now run ch5p3 in Appendix B. You will learn how to use MATLAB to convert a transfer function to state space in a specified form. The exercise solves the previous example by representing the transfer function in Eq. (5.45) by the state-space representation in parallel form of Eq. (5.49).

If the denominator of the transfer function has repeated real roots, the parallel form can still be derived from a partial-fraction expansion. However, the system matrix will not be diagonal. For example, assume the system

$$\frac{C(s)}{R(s)} = \frac{(s+3)}{(s+1)^2(s+2)} \tag{5.51}$$

which can be expanded as partial fractions:

$$\frac{C(s)}{R(s)} = \frac{2}{(s+1)^2} - \frac{1}{(s+1)} + \frac{1}{(s+2)} \tag{5.52}$$

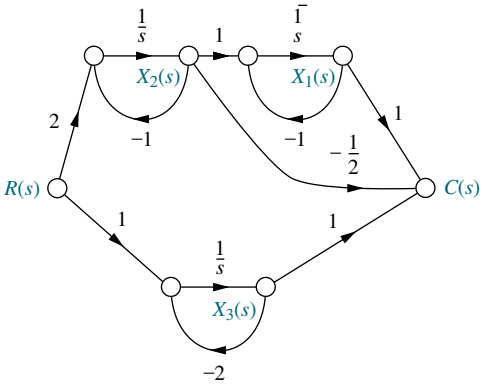


FIGURE 5.26 Signal-flow representation of Eq. (5.52)

Proceeding as before, the signal-flow graph for Eq. (5.52) is shown in Figure 5.26. The term $-1/(s+1)$ was formed by creating the signal flow from $X_2(s)$ to $C(s)$. Now the state and output equations can be written by inspection from Figure 5.26 as follows:

$$\dot{x}_1 = -x_1 + x_2 \tag{5.53a}$$

$$\dot{x}_2 = - x_2 + 2r \tag{5.53b}$$

$$\dot{x}_3 = - 2x_3 + r \tag{5.53c}$$

$$y = c(t) = x_1 - \frac{1}{2}x_2 + x_3 \tag{5.53d}$$

or, in vector-matrix form,

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} r \tag{5.54a}$$

$$y = \begin{bmatrix} 1 & -\frac{1}{2} & 1 \end{bmatrix} \mathbf{x} \tag{5.54b}$$

This system matrix, although not diagonal, has the system poles along the diagonal. Notice the 1 off the diagonal for the case of the repeated root. The form of the system matrix is known as the *Jordan canonical form.*

Controller Canonical Form

Another representation that uses phase variables is called the *controller canonical form*, so named for its use in the design of controllers, which is covered in Chapter 12. This form is obtained from the phase-variable form simply by ordering the phase variables in the reverse order. For example, consider the transfer function

$$G(s) = \frac{C(s)}{R(s)} = \frac{s^2 + 7s + 2}{s^3 + 9s^2 + 26s + 24} \tag{5.55}$$

The phase-variable form was derived in Example 3.5 as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -24 & -26 & -9\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}r \tag{5.56a}$$

$$y=\begin{bmatrix}2 & 7 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \tag{5.56b}$$

where $y = c(t)$. Renumbering the phase variables in reverse order yields

$$\begin{bmatrix}\dot{x}_3\\ \dot{x}_2\\ \dot{x}_1\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -24 & -26 & -9\end{bmatrix}\begin{bmatrix}x_3\\ x_2\\ x_1\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}r \tag{5.57a}$$

$$y=\begin{bmatrix}2 & 7 & 1\end{bmatrix}\begin{bmatrix}x_3\\ x_2\\ x_1\end{bmatrix} \tag{5.57b}$$

Finally, rearranging Eqs. (5.57) in ascending numerical order yields the controller canonical form[3] as

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}-9 & -26 & -24\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}1\\ 0\\ 0\end{bmatrix}r \tag{5.58a}$$

$$y=\begin{bmatrix}1 & 7 & 2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \tag{5.58b}$$

TryIt 5.3

Use the following MATLAB and Control System Toolbox statements to convert the transfer function of Eq. (5.55) to the controller canonical state-space representation of Eqs. (5.58).

```
numg=[1 7 2];
deng=[1 9 26 24];
[Acc,Bcc,Ccc,Dcc]...
 =tf2ss(numg,deng)
```

Figure 5.27 shows the steps we have taken on a signal-flow graph. Notice that the controller canonical form is obtained simply by renumbering the phase variables in the opposite order. Equations (5.56) can be obtained from Figure 5.27(*a*), and Eqs. (5.58) from Figure 5.27(*b*).

Notice that the phase-variable form and the controller canonical form contain the coefficients of the characteristic polynomial in the bottom row and in the top row,

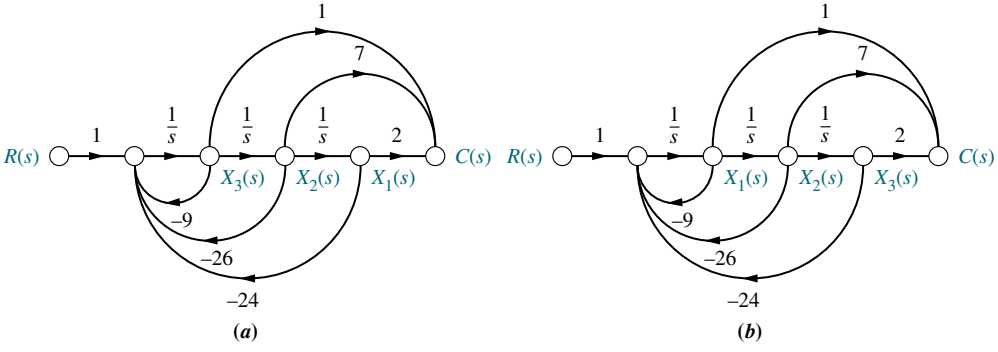


FIGURE 5.27 Signal-flow graphs for obtaining forms for $G(s) = C(s)/R(s) = (s^2 + 7s + 2)/(s^3 + 9s^2 + 26s + 24)$: **a.** phase-variable form; **b.** controller canonical form

[3] Students who are using MATLAB to convert from transfer functions to state space using the command **tf2ss** will notice that MATLAB reports the results in controller canonical form.

respectively. System matrices that contain the coefficients of the characteristic polynomial are called *companion matrices* to the characteristic polynomial. The phase-variable and controller canonical forms result in a lower and an upper companion system matrix, respectively. Companion matrices can also have the coefficients of the characteristic polynomial in the left or right column. In the next subsection, we discuss one of these representations.

Observer Canonical Form

The *observer canonical form*, so named for its use in the design of observers (covered in Chapter 12), is a representation that yields a left companion system matrix. As an example, the system modeled by Eq. (5.55) will be represented in this form. Begin by dividing all terms in the numerator and denominator by the highest power of s, s^3, and obtain

$$\frac{C(s)}{R(s)} = \frac{\frac{1}{s} + \frac{7}{s^2} + \frac{2}{s^3}}{1 + \frac{9}{s} + \frac{26}{s^2} + \frac{24}{s^3}} \tag{5.59}$$

Cross-multiplying yields

$$\left[\frac{1}{s} + \frac{7}{s^2} + \frac{2}{s^3}\right] R(s) = \left[1 + \frac{9}{s} + \frac{26}{s^2} + \frac{24}{s^3}\right] C(s) \tag{5.60}$$

Combining terms of like powers of integration gives

$$C(s) = \frac{1}{s}[R(s) - 9C(s)] + \frac{1}{s^2}[7R(s) - 26C(s)] + \frac{1}{s^3}[2R(s) - 24C(s)] \tag{5.61}$$

or

$$C(s) = \frac{1}{s}\left[[R(s) - 9C(s)] + \frac{1}{s}\left([7R(s) - 26C(s)] + \frac{1}{s}[2R(s) - 24C(s)]\right)\right] \tag{5.62}$$

Equation (5.61) or (5.62) can be used to draw the signal-flow graph. Start with three integrations, as shown in Figure 5.28(*a*).

Using Eq. (5.61), the first term tells us that output $C(s)$ is formed, in part, by integrating $[R(s) - 9C(s)]$. We thus form $[R(s) - 9C(s)]$ at the input to the integrator closest to the output, $C(s)$, as shown in Figure 5.28(*b*). The second term tells us that the

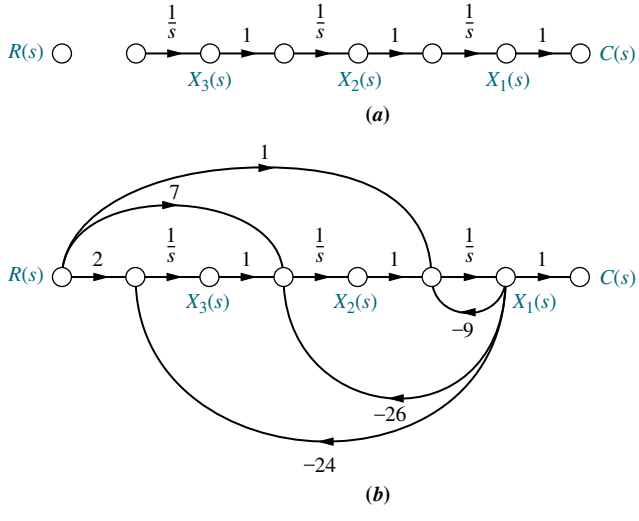


FIGURE 5.28 Signal-flow graph for observer canonical form variables: **a.** planning; **b.** implementation

term $[7R(s) - 26C(s)]$ must be integrated twice. Now form $[7R(s) - 26C(s)]$ at the input to the second integrator. Finally, the last term of Eq. (5.61) says $[2R(s) - 24C(s)]$ must be integrated three times. Form $[2R(s) - 24C(s)]$ at the input to the first integrator.

Identifying the state variables as the outputs of the integrators, we write the following state equations:

$$\dot{x}_1 = -9x_1 + x_2 + r \tag{5.63a}$$

$$\dot{x}_2 = -26x_1 + x_3 + 7r \tag{5.63b}$$

$$\dot{x}_3 = -24x_1 + 2r \tag{5.63c}$$

The output equation from Figure 5.28(b) is

$$y = c(t) = x_1 \tag{5.64}$$

In vector-matrix form, Eqs. (5.63) and (5.64) become

$$\dot{\mathbf{x}} = \begin{bmatrix} -9 & 1 & 0 \\ -26 & 0 & 1 \\ -24 & 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 7 \\ 2 \end{bmatrix} r \tag{5.65a}$$

$$y = [1 \quad 0 \quad 0]\mathbf{x} \tag{5.65b}$$

Notice that the form of Eqs. (5.65) is similar to the phase-variable form, except that the coefficients of the denominator of the transfer function are in the first column, and the coefficients of the numerator form the input matrix, **B**. Also notice that the observer canonical form has an **A** matrix that is the transpose of the controller canonical form, a **B** vector that is the transpose of the controller canonical form's **C** vector, and a **C** vector that is the transpose of the controller canonical form's **B** vector. We therefore say that these two forms are *duals*. Thus, if a system is described by **A, B,** and **C**, its dual is described by $\mathbf{A_D} = \mathbf{A}^T$, $\mathbf{B_D} = \mathbf{C}^T$, $\mathbf{C_D} = \mathbf{B}^T$. You can verify the significance of duality by comparing the signal-flow graphs of a system and its dual, Figures 5.27(b) and 5.28(b), respectively. The signal-flow graph of the dual can be obtained from that of the original by reversing all arrows, changing state variables to their derivatives and vice versa, and interchanging $C(s)$ and $R(s)$, thus reversing the roles of the input and the output.

We conclude this section with an example that demonstrates the application of the previously discussed forms to a feedback control system.

TryIt 5.4

Use the following MATLAB and Control System Toolbox statements to convert the transfer function of Eq. (5.55) to the observer canonical state-space representation of Eqs. (5.65).

```
numg=[1 7 2];
deng=[1 9 26 24];
[Acc,Bcc,Ccc,Dcc]...
 =tf2ss(numg,deng);
Aoc=transpose(Acc)
Boc=transpose(Ccc)
Coc=transpose(Bcc)
```

Example 5.8

State-Space Representation of Feedback Systems

PROBLEM: Represent the feedback control system shown in Figure 5.29 in state space. Model the forward transfer function in cascade form.

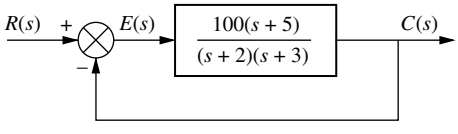


FIGURE 5.29 Feedback control system for Example 5.8

SOLUTION: First we model the forward transfer function in cascade form. The gain of 100, the pole at -2, and the pole at -3 are shown cascaded in Figure 5.30(a). The zero at -5 was obtained using the method for implementing zeros for a system represented in phase-variable form, as discussed in Section 3.5.

Next add the feedback and input paths, as shown in Figure 5.30(b). Now, by inspection, write the state equations:

$$\dot{x}_1 = -3x_1 + x_2 \tag{5.66a}$$

$$\dot{x}_2 = -2x_2 + 100(r - c) \tag{5.66b}$$

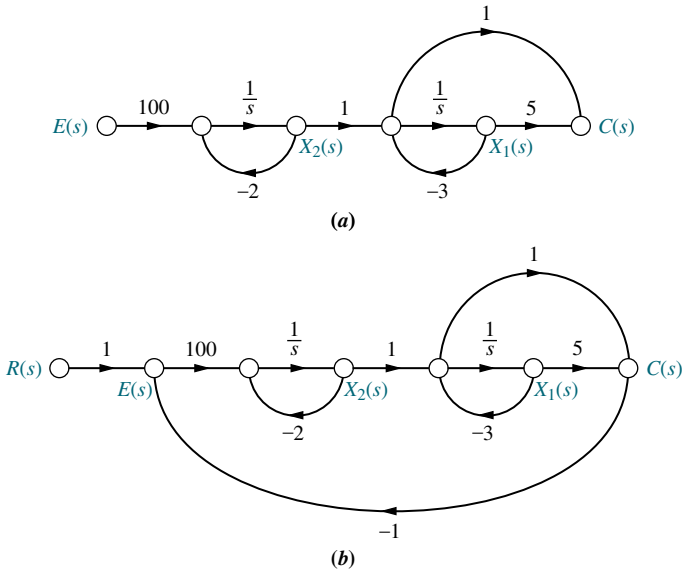


FIGURE 5.30 Creating a signal-flow graph for the Figure 5.29 system: **a.** forward transfer function; **b.** complete system

But, from Figure 5.30*(b)*,

$$c = 5x_1 + (x_2 - 3x_1) = 2x_1 + x_2 \tag{5.67}$$

Substituting Eq. (5.67) into (5.66b), we find the state equations for the system:

$$\dot{x}_1 = \quad -3x_1 \quad + x_2 \tag{5.68a}$$

$$\dot{x}_2 = -200x_1 - 102x_2 + 100r \tag{5.68b}$$

The output equation is the same as Eq. (5.67), or

$$y = c(t) = 2x_1 + x_2 \tag{5.69}$$

In vector-matrix form

$$\dot{\mathbf{x}} = \begin{bmatrix} -3 & 1 \\ -200 & -102 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 100 \end{bmatrix} r \tag{5.70a}$$

$$y = [2 \quad 1]\mathbf{x} \tag{5.70b}$$

Skill-Assessment Exercise 5.6

WileyPLUS
WPCS
Control Solutions

PROBLEM: Represent the feedback control system shown in Figure 5.29 in state space. Model the forward transfer function in controller canonical form.

ANSWER:

$$\dot{\mathbf{x}} = \begin{bmatrix} -105 & -506 \\ 1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r$$

$$y = [100 \quad 500]\mathbf{x}$$

The complete solution is at www.wiley.com/college/nise.

In this section, we used transfer functions and signal-flow graphs to represent systems in parallel, cascade, controller canonical, and observer canonical forms, in addition to the phase-variable form. Using the transfer function $C(s)/R(s) = (s+3)/[(s+4)(s+6)]$ as an example, Figure 5.31 compares the aforementioned forms. Notice the duality of the controller and observer canonical forms, as demonstrated by their respective signal-flow graphs and state equations. In the next section, we will explore the possibility of transforming between representations without using transfer functions and signal-flow graphs.

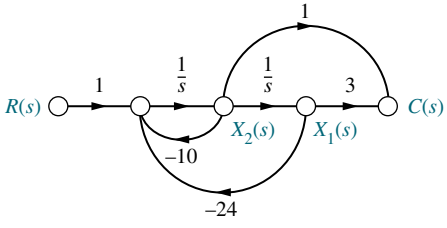
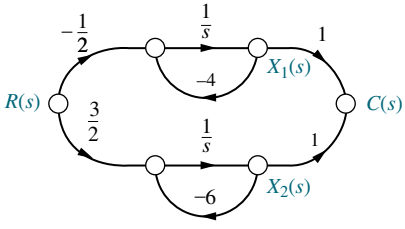
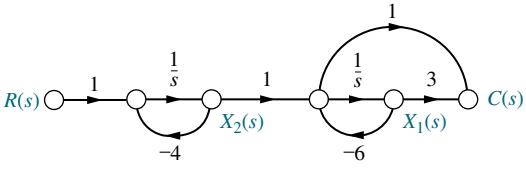
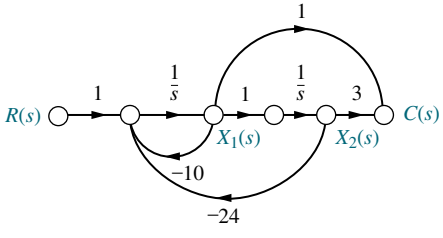
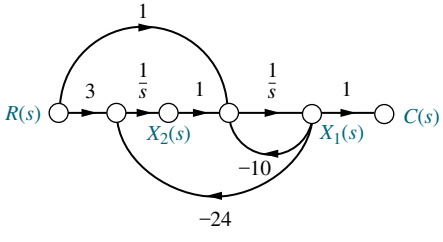
Form	Transfer function	Signal-flow diagram	State equations
Phase variable	$\frac{1}{(s^2+10s+24)} * (s+3)$		$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -24 & -10 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r$ $y = [3 \;\; 1]\,\mathbf{x}$
Parallel	$\frac{-1/2}{(s+4)} + \frac{3/2}{s+6}$		$\dot{\mathbf{x}} = \begin{bmatrix} -4 & 0 \\ 0 & -6 \end{bmatrix}\mathbf{x} + \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \end{bmatrix} r$ $y = [1 \;\; 1]\,\mathbf{x}$
Cascade	$\frac{1}{(s+4)} * \frac{(s+3)}{(s+6)}$		$\dot{\mathbf{x}} = \begin{bmatrix} -6 & 1 \\ 0 & -4 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r$ $y = [-3 \;\; 1]\,\mathbf{x}$
Controller canonical	$\frac{1}{(s^2+10s+24)} * (s+3)$		$\dot{\mathbf{x}} = \begin{bmatrix} -10 & -24 \\ 1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r$ $y = [1 \;\; 3]\,\mathbf{x}$
Observer canonical	$\frac{\frac{1}{s} + \frac{3}{s^2}}{1 + \frac{10}{s} + \frac{24}{s^2}}$		$\dot{\mathbf{x}} = \begin{bmatrix} -10 & 1 \\ -24 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 3 \end{bmatrix} r$ $y = [1 \;\; 0]\,\mathbf{x}$

FIGURE 5.31 State-space forms for $C(s)/R(s) = (s+3)/[(s+4)(s+6)]$. Note: $y = c(t)$

5.8 Similarity Transformations

State Space SS

In Section 5.7, we saw that systems can be represented with different state variables even though the transfer function relating the output to the input remains the same. The various forms of the state equations were found by manipulating the transfer function, drawing a signal-flow graph, and then writing the state equations from the signal-flow graph. These systems are called *similar systems*. Although their state-space representations are different, similar systems have the same transfer function and hence the same poles and eigenvalues.

We can make transformations between similar systems from one set of state equations to another without using the transfer function and signal-flow graphs. The results are presented in this section along with examples. Students who have not broached this subject in the past or who wish to refresh their memories are encouraged to study Appendix L at www.wiley.com/college/nise for the derivation. The result of the derivation states: A system represented in state space as

$$\boxed{\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}} \tag{5.71a}$$

$$\boxed{\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}} \tag{5.71b}$$

can be transformed to a similar system,

$$\boxed{\mathbf{z} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{z} + \mathbf{P}^{-1}\mathbf{B}\mathbf{u}} \tag{5.72a}$$

$$\boxed{\mathbf{y} = \mathbf{C}\mathbf{P}\mathbf{z} + \mathbf{D}\mathbf{u}} \tag{5.72b}$$

where, for 2-space,

$$\boxed{\mathbf{P} = [\mathbf{U}_{z_1}\,\mathbf{U}_{z_2}] = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}} \tag{5.72c}$$

$$\boxed{\mathbf{x} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathbf{P}\mathbf{z}} \tag{5.72d}$$

and

$$\boxed{\mathbf{z} = \mathbf{P}^{-1}\mathbf{x}} \tag{5.72e}$$

Thus, $\mathbf{P}$ is a transformation matrix whose columns are the coordinates of the basis vectors of the z_1z_2 space expressed as linear combinations of the x_1x_2 space. Let us look at an example.

Example 5.9

Similarity Transformations on State Equations

PROBLEM: Given the system represented in state space by Eqs. (5.73),

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -7 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{5.73a}$$

$$y = [1 \quad 0 \quad 0]\mathbf{x} \tag{5.73b}$$

transform the system to a new set of state variables, $\mathbf{z}$, where the new state variables are related to the original state variables, $\mathbf{x}$, as follows:

$$z_1 = 2x_1 \tag{5.74a}$$

$$z_2 = 3x_1 + 2x_2 \tag{5.74b}$$

$$z_3 = x_1 + 4x_2 + 5x_3 \tag{5.74c}$$

SOLUTION: Expressing Eqs. (5.74) in vector-matrix form,

$$\mathbf{z} = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 2 & 0 \\ 1 & 4 & 5 \end{bmatrix} \mathbf{x} = \mathbf{P}^{-1}\mathbf{x} \tag{5.75}$$

Using Eqs. (5.72) as a guide,

$$\begin{aligned} \mathbf{P}^{-1}\mathbf{A}\mathbf{P} &= \begin{bmatrix} 2 & 0 & 0 \\ 3 & 2 & 0 \\ 1 & 4 & 5 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -7 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ -0.75 & 0.5 & 0 \\ 0.5 & -0.4 & 0.2 \end{bmatrix} \\ &= \begin{bmatrix} -1.5 & 1 & 0 \\ -1.25 & 0.7 & 0.4 \\ -2.5 & 0.4 & -6.2 \end{bmatrix} \end{aligned} \tag{5.76}$$

$$\mathbf{P}^{-1}\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 2 & 0 \\ 1 & 4 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \tag{5.77}$$

$$\mathbf{C}\mathbf{P} = [1 \quad 0 \quad 0] \begin{bmatrix} 0.5 & 0 & 0 \\ -0.75 & 0.5 & 0 \\ 0.5 & -0.4 & 0.2 \end{bmatrix} = [0.5 \quad 0 \quad 0] \tag{5.78}$$

Therefore, the transformed system is

$$\dot{\mathbf{z}} = \begin{bmatrix} -1.5 & 1 & 0 \\ -1.25 & 0.7 & 0.4 \\ -2.55 & 0.4 & -6.2 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} u \tag{5.79a}$$

$$y = [0.5 \quad 0 \quad 0]\mathbf{z} \tag{5.79b}$$

MATLAB

ML

Students who are using MATLAB should now run ch5p4 in Appendix B. You will learn how to perform similarity transformations. This exercise uses MATLAB to do Example 5.9.

Thus far we have talked about transforming systems between basis vectors in a different state space. One major advantage of finding these similar systems is apparent in the transformation to a system that has a diagonal matrix.

Diagonalizing a System Matrix

In Section 5.7, we saw that the parallel form of a signal-flow graph can yield a diagonal system matrix. A diagonal system matrix has the advantage that each state equation is a function of only one state variable. Hence, each differential equation can be solved independently of the other equations. We say that the equations are *decoupled*.

Rather than using partial fraction expansion and signal-flow graphs, we can decouple a system using matrix transformations. If we find the correct matrix, $\mathbf{P}$, the transformed system matrix, $\mathbf{P}^{-1}\mathbf{AP}$, will be a diagonal matrix. Thus, we are looking for a transformation to another state space that yields a diagonal matrix in that space. This new state space also has basis vectors that lie along its state variables. We give a special name to any vectors that are collinear with the basis vectors of the new system that yields a diagonal system matrix: they are called *eigenvectors*. Thus, the coordinates of the eigenvectors form the columns of the transformation matrix, $\mathbf{P}$, as we demonstrate in Eq. L.7 in Appendix L at www.wiley.com/college/nise.

First, let us formally define eigenvectors from another perspective and then show that they have the property just described. Then we will define eigenvalues. Finally, we will show how to diagonalize a matrix.

Definitions

Eigenvector. The eigenvectors of the matrix $\mathbf{A}$ are all vectors, $\mathbf{x_i} \neq \mathbf{0}$, which under the transformation $\mathbf{A}$ become multiples of themselves; that is,

$$\boxed{\mathbf{Ax_i} = \lambda_i \mathbf{x_i}} \tag{5.80}$$

where λ_i's are constants.

Figure 5.32 shows this definition of eigenvectors. If $\mathbf{Ax}$ is not collinear with $\mathbf{x}$ after the transformation, as in Figure 5.32(a), $\mathbf{x}$ is not an eigenvector. If $\mathbf{Ax}$ is collinear with $\mathbf{x}$ after the transformation, as in Figure 5.32(b), $\mathbf{x}$ is an eigenvector.

Eigenvalue. The eigenvalues of the matrix $\mathbf{A}$ are the values of λ_i that satisfy Eq. (5.80) for $\mathbf{x_i} \neq \mathbf{0}$.

To find the eigenvectors, we rearrange Eq. (5.80). Eigenvectors, $\mathbf{x_i}$, satisfy

$$\mathbf{0} = (\lambda_i \mathbf{I} - \mathbf{A})\mathbf{x_i} \tag{5.81}$$

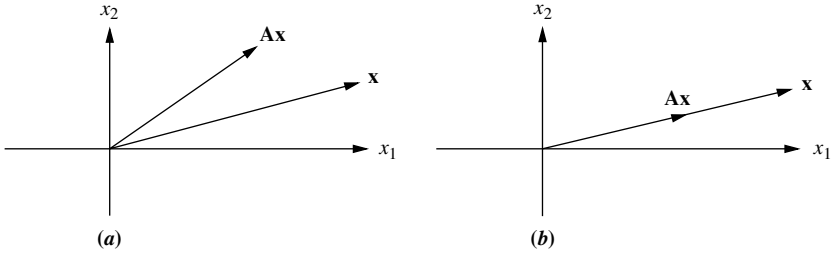


FIGURE 5.32 To be an eigenvector, the transformation $\mathbf{Ax}$ must be collinear with $\mathbf{x}$; thus, in ($\boldsymbol{a}$), $\mathbf{x}$ is not an eigenvector; in ($\boldsymbol{b}$), it is

Solving for $\mathbf{x_i}$ by premultiplying both sides by $(\lambda_i\mathbf{I}-\mathbf{A})^{-1}$ yields

$$\mathbf{x_i} = (\lambda_i\mathbf{I}-\mathbf{A})^{-1}\mathbf{0} = \frac{\text{adj}(\lambda_i\mathbf{I}-\mathbf{A})}{\det(\lambda_i\mathbf{I}-\mathbf{A})}\mathbf{0} \tag{5.82}$$

Since $\mathbf{x_i} \neq \mathbf{0}$, a nonzero solution exists if

$$\boxed{\det(\lambda_i\mathbf{I}-\mathbf{A}) = \mathbf{0}} \tag{5.83}$$

from which λ_i, the eigenvalues, can be found.

We are now ready to show how to find the eigenvectors, $\mathbf{x_i}$. First we find the eigenvalues, λ_i, using $\det(\lambda_i\mathbf{I}-\mathbf{A}) = \mathbf{0}$, and then we use Eq. (5.80) to find the eigenvectors.

Example 5.10

Finding Eigenvectors

PROBLEM: Find the eigenvectors of the matrix

$$\mathbf{A} = \begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix} \tag{5.84}$$

SOLUTION: The eigenvectors, $\mathbf{x_i}$, satisfy Eq. (5.81). First, use $\det(\lambda_i\mathbf{I}-\mathbf{A}) = 0$ to find the eigenvalues, λ_i, for Eq. (5.81):

$$\begin{aligned} \det(\lambda\mathbf{I}-\mathbf{A}) &= \left| \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix} \right| \\ &= \begin{vmatrix} \lambda+3 & -1 \\ -1 & \lambda+3 \end{vmatrix} \\ &= \lambda^2 + 6\lambda + 8 \end{aligned} \tag{5.85}$$

from which the eigenvalues are $\lambda = -2$, and -4.

Using Eq. (5.80) successively with each eigenvalue, we have

$$\mathbf{A}\mathbf{x_i} = \lambda\mathbf{x_i}$$

$$\begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5.86}$$

or

$$-3x_1 + x_2 = -2x_1 \tag{5.87a}$$

$$x_1 - 3x_2 = -2x_2 \tag{5.87b}$$

from which $x_1 = x_2$. Thus,

$$\mathbf{x} = \begin{bmatrix} c \\ c \end{bmatrix} \tag{5.88}$$

Using the other eigenvalue, -4, we have

$$\mathbf{x} = \begin{bmatrix} c \\ -c \end{bmatrix} \tag{5.89}$$

Using Eqs. (5.88) and (5.89), one choice of eigenvectors is

$$\mathbf{x_1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{x_2} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{5.90}$$

We now show that if the eigenvectors of the matrix **A** are chosen as the basis vectors of a transformation, **P**, the resulting system matrix will be diagonal. Let the transformation matrix **P** consist of the eigenvectors of **A,** $\mathbf{x_i}$.

$$\mathbf{P} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x_n}] \tag{5.91}$$

Since $\mathbf{x_i}$ are eigenvectors, $\mathbf{A}\mathbf{x_i} = \lambda_i \mathbf{x_i}$, which can be written equivalently as a set of equations expressed by

$$\mathbf{AP} = \mathbf{PD} \tag{5.92}$$

where **D** is a diagonal matrix consisting of λ_i's, the eigenvalues, along the diagonal, and **P** is as defined in Eq. (5.91). Solving Eq. (5.92) for **D** by premultiplying by $\mathbf{P}^{-1}$, we get

$$\mathbf{D} = \mathbf{P}^{-1}\mathbf{AP} \tag{5.93}$$

which is the system matrix of Eq. (5.72).

In summary, under the transformation **P**, consisting of the eigenvectors of the system matrix, the transformed system is diagonal, with the eigenvalues of the system along the diagonal. The transformed system is identical to that obtained using partial-fraction expansion of the transfer function with distinct real roots.

In Example 5.10, we found eigenvectors of a second-order system. Let us continue with this problem and diagonalize the system matrix.

Example 5.11

Diagonalizing a System in State Space

PROBLEM: Given the system of Eqs. (5.94), find the diagonal system that is similar.

$$\dot{\mathbf{x}} = \begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u \tag{5.94a}$$

$$y = [2 \quad 3]\mathbf{x} \tag{5.94b}$$

SOLUTION: First find the eigenvalues and the eigenvectors. This step was performed in Example 5.10. Next form the transformation matrix **P**, whose columns consist of the eigenvectors.

$$\mathbf{P} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{5.95}$$

Finally, form the similar system's system matrix, input matrix, and output matrix, respectively.

$$\mathbf{P}^{-1}\mathbf{AP} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}\begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -4 \end{bmatrix} \tag{5.96a}$$

$$\mathbf{P}^{-1}\mathbf{B} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3/2 \\ -1/2 \end{bmatrix} \tag{5.96b}$$

$$\mathbf{CP} = [2 \quad 3]\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [5 \quad -1] \tag{5.96c}$$

Substituting Eqs. (5.96) into Eqs. (5.72), we get

$$\dot{\mathbf{z}} = \begin{bmatrix} -2 & 0 \\ 0 & -4 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 3/2 \\ -1/2 \end{bmatrix}u \tag{5.97a}$$

$$y = [5 \quad -1]\mathbf{z} \tag{5.97b}$$

Notice that the system matrix is diagonal, with the eigenvalues along the diagonal.

MATLAB ML

Students who are using MATLAB should now run ch5p5 in Appendix B. This problem, which uses MATLAB to diagonalize a system, is similar (but not identical) to Example 5.11.

Skill-Assessment Exercise 5.7

WileyPLUS WPCS Control Solutions

PROBLEM: For the system represented in state space as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & 3 \\ -4 & -6 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 3 \end{bmatrix}u$$

$$y = [1 \quad 4]\mathbf{x}$$

convert the system to one where the new state vector, $\mathbf{z}$, is

$$\mathbf{z} = \begin{bmatrix} 3 & -2 \\ 1 & -4 \end{bmatrix}\mathbf{x}$$

ANSWER:

$$\dot{\mathbf{z}} = \begin{bmatrix} 6.5 & -8.5 \\ 9.5 & -11.5 \end{bmatrix}\mathbf{z} + \begin{bmatrix} -3 \\ -11 \end{bmatrix}u$$

$$y = [0.8 \quad -1.4]\mathbf{z}$$

The complete solution is at www.wiley.com/college/nise.

Skill-Assessment Exercise 5.8

PROBLEM: For the original system of Skill-Assessment Exercise 5.7, find the diagonal system that is similar.

ANSWER:

$$\dot{\mathbf{z}} = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 18.39 \\ 20 \end{bmatrix}u$$

$$y = [-2.121 \quad 2.6]\mathbf{z}$$

The complete solution is at www.wiley.com/college/nise.

TryIt 5.5

Use the following MATLAB and Control System Toolbox statements to do Skill-Assessment Exercise 5.8.

```
A=[1 3;-4 -6];
B=[1;3];
C=[1 4];
D=0;S=ss(A,B,C,D);
Sd=canon(S,'modal')
```

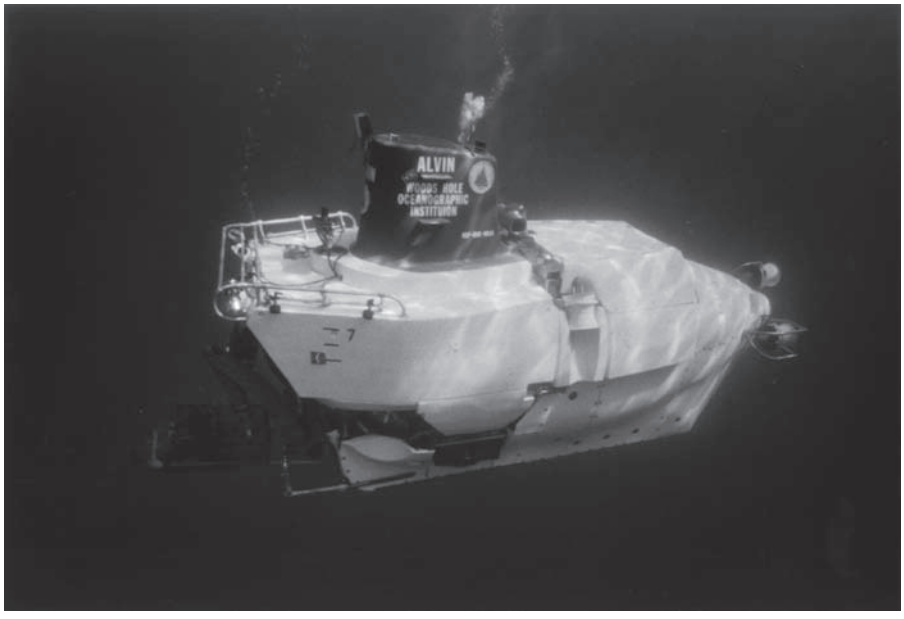


FIGURE 5.33 *Alvin*, a manned submersible, explored the wreckage of the *Titanic* with a tethered robot, *Jason Junior*.

In this section, we learned how to move between different state-space representations of the same system via matrix transformations rather than transfer function manipulation and signal-flow graphs. These different representations are called *similar*. The characteristics of similar systems are that the transfer functions relating the output to the input are the same, as are the eigenvalues and poles. A particularly useful transformation was converting a system with distinct, real eigenvalues to a diagonal system matrix.

We now summarize the concepts of block diagram and signal-flow representations of systems, first through case study problems and then in a written summary. Our case studies include the antenna azimuth position control system and the Unmanned Free-Swimming Submersible vehicle (UFSS). Block diagram reduction is important for the analysis and design of these systems as well as the control systems on board *Alvin* (Figure 5.33), used to explore the wreckage of the *Titanic* 13,000 feet under the Atlantic in July 1986 (*Ballard*, *1987*).

Case Studies

Antenna Control: Designing a Closed-Loop Response

Design
D

This chapter has shown that physical subsystems can be modeled mathematically with transfer functions and then interconnected to form a feedback system. The interconnected mathematical models can be reduced to a single transfer function representing the system from input to output. This transfer function, the closed-loop transfer function, is then used to determine the system response.

The following case study shows how to reduce the subsystems of the antenna azimuth position control system to a single, closed-loop transfer function in order to analyze and design the transient response characteristics.

PROBLEM: Given the antenna azimuth position control system shown on the front endpapers, Configuration 1, do the following:

a. Find the closed-loop transfer function using block diagram reduction.

b. Represent each subsystem with a signal-flow graph and find the state-space representation of the closed-loop system from the signal-flow graph.

c. Use the signal-flow graph found in **b** along with Mason's rule to find the closed-loop transfer function.

d. Replace the power amplifier with a transfer function of unity and evaluate the closed-loop peak time, percent overshoot, and settling time for $K = 1000$.

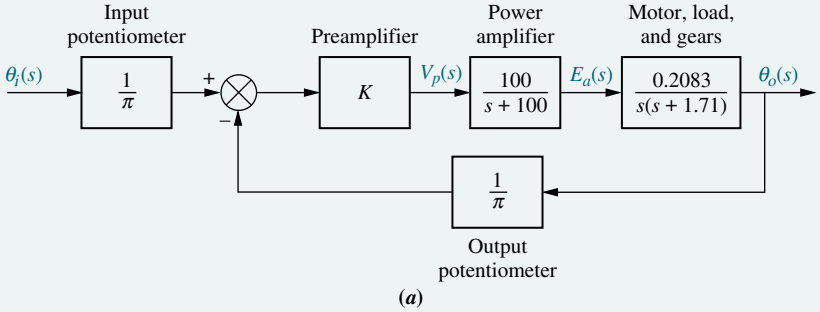
e. For the system of **d**, derive the expression for the closed-loop step response of the system.

f. For the simplified model of **d**, find the value of K that yields a 10% overshoot.

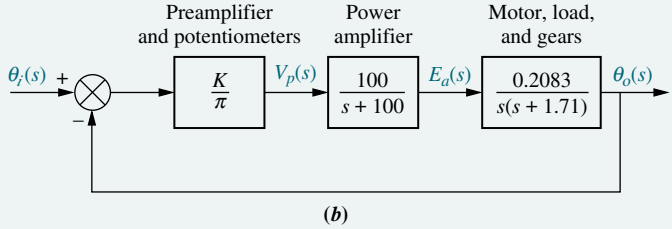
State Space
SS

SOLUTION: Each subsystem's transfer function was evaluated in the case study in Chapter 2. We first assemble them into the closed-loop, feedback control system block diagram shown in Figure 5.34(a).

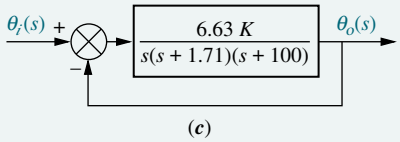
a. The steps taken to reduce the block diagram to a single, closed-loop transfer function relating the output angular displacement to the input angular displacement are shown in Figure 5.34(a–d). In Figure 5.34(b), the input potentiometer was pushed to the right past the summing junction, creating a unity feedback



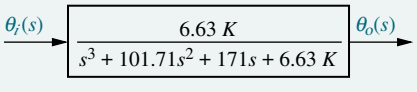
(*a*)



(*b*)



(*c*)



(*d*)

FIGURE 5.34 Block diagram reduction for the antenna azimuth position control system: **a.** original; **b.** pushing input potentiometer to the right past the summing junction; **c.** showing equivalent forward transfer function; **d.** final closed-loop transfer function

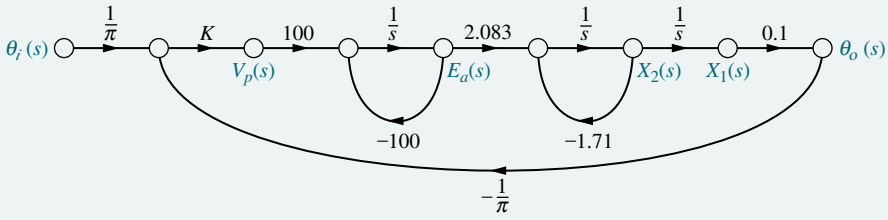


FIGURE 5.35 Signal-flow graph for the antenna azimuth position control system

system. In Figure 5.34(c), all the blocks of the forward transfer function are multiplied together, forming the equivalent forward transfer function. Finally, the feedback formula is applied, yielding the closed-loop transfer function in Figure 5.34(d).

State Space
SS

b. In order to obtain the signal-flow graph of each subsystem, we use the state equations derived in the case study of Chapter 3. The signal-flow graph for the power amplifier is drawn from the state equations of Eqs. (3.87) and (3.88), and the signal-flow graph of the motor and load is drawn from the state equation of Eq. (3.98). Other subsystems are pure gains. The signal-flow graph for Figure 5.34(a) is shown in Figure 5.35 and consists of the interconnected subsystems.

The state equations are written from Figure 5.35. First define the state variables as the outputs of the integrators. Hence, the state vector is

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ e_a \end{bmatrix} \tag{5.98}$$

Using Figure 5.35, we write the state equations by inspection:

$$\dot{x}_1 = +x_2 \tag{5.99a}$$

$$\dot{x}_2 = -1.71x_2 + 2.083e_a \tag{5.99b}$$

$$\dot{e}_a = -3.18Kx_1 - 100e_a + 31.8K\theta_i \tag{5.99c}$$

along with the output equation,

$$y = \theta_o = 0.1x_1 \tag{5.100}$$

where $1/\pi = 0.318$.

In vector-matrix form,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1.71 & 2.083 \\ -3.18K & 0 & -100 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 31.8K \end{bmatrix} \theta_i \tag{5.101a}$$

$$y = [0.1 \quad 0 \quad 0]\mathbf{x} \tag{5.101b}$$

c. We now apply Mason's rule to Figure 5.35 to derive the closed-loop transfer function of the antenna azimuth position control system. First find the forward-path gains. From Figure 5.35 there is only one forward-path gain:

$$T_1 = \left(\frac{1}{\pi}\right)(K)(100)\left(\frac{1}{s}\right)(2.083)\left(\frac{1}{s}\right)\left(\frac{1}{s}\right)(0.1) = \frac{6.63K}{s^3} \tag{5.102}$$

Next identify the closed-loop gains. There are three: the power amplifier loop, $G_{L1}(s)$, with e_a at the output; the motor loop, $G_{L2}(s)$, with x_2 at the output; and the entire system loop, $G_{L3}(s)$, with θ_0 at the output.

$$G_{L1}(s) = \frac{-100}{s} \tag{5.103a}$$

$$G_{L2}(s) = \frac{-1.71}{s} \tag{5.103b}$$

$$G_{L3}(s) = (K)(100)\left(\frac{1}{s}\right)(2.083)\left(\frac{1}{s}\right)\left(\frac{1}{s}\right)(0.1)\left(\frac{-1}{\pi}\right) = \frac{-6.63K}{s^3} \tag{5.103c}$$

Only $G_{L1}(s)$ and $G_{L2}(s)$ are nontouching loops. Thus, the nontouching-loop gain is

$$G_{L1}(s)G_{L2}(s) = \frac{171}{s^2} \tag{5.104}$$

Forming Δ and Δ_k in Eq. (5.28), we have

$$\begin{aligned}\Delta &= 1 - [G_{L1}(s) + G_{L2}(s) + G_{L3}(s)] + [G_{L1}(s)G_{L2}(s)] \\ &= 1 + \frac{100}{s} + \frac{1.71}{s} + \frac{6.63K}{s^3} + \frac{171}{s^2}\end{aligned} \tag{5.105}$$

and

$$\Delta_1 = 1 \tag{5.106}$$

Substituting Eqs. (5.102), (5.105), and (5.106) into Eq. (5.28), we obtain the closed-loop transfer function as

$$T(s) = \frac{C(s)}{R(s)} = \frac{T_1\Delta_1}{\Delta} = \frac{6.63K}{s^3 + 101.71s^2 + 171s + 6.63K} \tag{5.107}$$

d. Replacing the power amplifier with unity gain and letting the preamplifier gain, K, in Figure 5.34(b) equal 1,000 yield a forward transfer function, $G(s)$, of

$$G(s) = \frac{66.3}{s(s + 1.71)} \tag{5.108}$$

Using the feedback formula to evaluate the closed-loop transfer function, we obtain

$$T(s) = \frac{66.3}{s^2 + 1.71s + 66.3} \tag{5.109}$$

From the denominator, $\omega_n = 8.14$, $\zeta = 0.105$. Using Eqs. (4.34), (4.38), and (4.42), the peak time = 0.388 second, the percent overshoot = 71.77%, and the settling time = 4.68 seconds.

e. The Laplace transform of the step response is found by first multiplying‘ Eq. (5.109) by $1/s$, a unit-step input, and expanding into partial fractions:

$$\begin{aligned}C(s) = \frac{66.3}{s(s^2 + 1.71s + 66.3)} &= \frac{1}{s} - \frac{s + 1.71}{s^2 + 1.71s + 66.3} \\ &= \frac{1}{s} - \frac{(s + 0.855) + 0.106(8.097)}{(s + 0.855)^2 + (8.097)^2}\end{aligned} \tag{5.110}$$

Taking the inverse Laplace transform, we find

$$c(t) = 1 - e^{-0.855t}(\cos 8.097t + 0.106 \sin 8.097t) \tag{5.111}$$

f. For the simplified model we have

$$G(s) = \frac{0.0663K}{s(s + 1.71)} \tag{5.112}$$

from which the closed-loop transfer function is calculated to be

$$T(s) = \frac{0.0663K}{s^2 + 1.71s + 0.0663K} \tag{5.113}$$

From Eq. (4.39) a 10% overshoot yields $\zeta = 0.591$. Using the denominator of Eq. (5.113), $\omega_n = \sqrt{0.0663K}$ and $2\zeta\omega_n = 1.71$. Thus,

$$\zeta = \frac{1.71}{2\sqrt{0.0663K}} = 0.591 \tag{5.114}$$

from which $K = 31.6$.

CHALLENGE: You are now given a problem to test your knowledge of this chapter's objectives: Referring to the antenna azimuth position control system shown on the front endpapers, Configuration 2, do the following:

State Space SS

a. Find the closed-loop transfer function using block diagram reduction.

b. Represent each subsystem with a signal-flow graph and find the state-space representation of the closed-loop system from the signal-flow graph.

c. Use the signal-flow graph found in (b) along with Mason's rule to find the closed-loop transfer function.

d. Replace the power amplifier with a transfer function of unity and evaluate the closed-loop percent overshoot, settling time, and peak time for $K = 5$.

e. For the system used for (d), derive the expression for the closed-loop step response.

f. For the simplified model in (d), find the value of preamplifier gain, $\boldsymbol{K}$, to yield 15% overshoot.

UFSS Vehicle: Pitch-Angle Control Representation

State Space SS

We return to the Unmanned Free-Swimming Submersible (UFSS) vehicle introduced in the case studies in Chapter 4 (*Johnson, 1980*). We will represent in state space the pitch-angle control system that is used for depth control.

PROBLEM: Consider the block diagram of the pitch control loop of the UFSS vehicle shown on the back endpapers. The pitch angle, θ, is controlled by a commanded pitch angle, θ_e, which along with pitch-angle and pitch-rate feedback determines the elevator deflection, δ_e, which acts through the vehicle dynamics to determine the pitch angle. Let $K_1 = K_2 = 1$ and do the following:

a. Draw the signal-flow graph for each subsystem, making sure that pitch angle, pitch rate, and elevator deflection are represented as state variables. Then interconnect the subsystems.

b. Use the signal-flow graph obtained in **a** to represent the pitch control loop in state space.

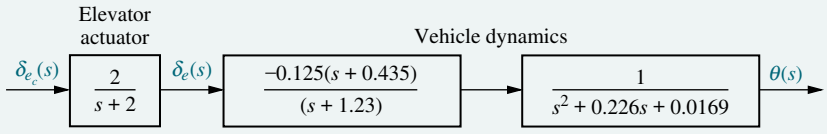


FIGURE 5.36 Block diagram of the UFSS vehicle's elevator and vehicle dynamics, from which a signal-flow graph can be drawn

SOLUTION:

a. The vehicle dynamics are split into two transfer functions, from which the signal-flow graph is drawn. Figure 5.36 shows the division along with the elevator actuator. Each block is drawn in phase-variable form to meet the requirement that particular system variables be state variables. This result is shown in Figure 5.37(a). The feedback paths are then added to complete the signal-flow graph, which is shown in Figure 5.37(b).

b. By inspection, the derivatives of state variables x_1 through x_4 are written as

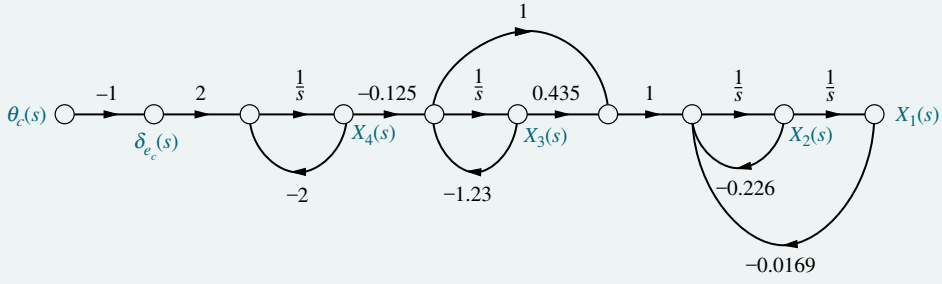
$$\dot{x}_1 = x_2 \tag{5.115a}$$

$$\dot{x}_2 = -0.0169x_1 - 0.226x_2 + 0.435x_3 - 1.23x_3 - 0.125x_4 \tag{5.115b}$$

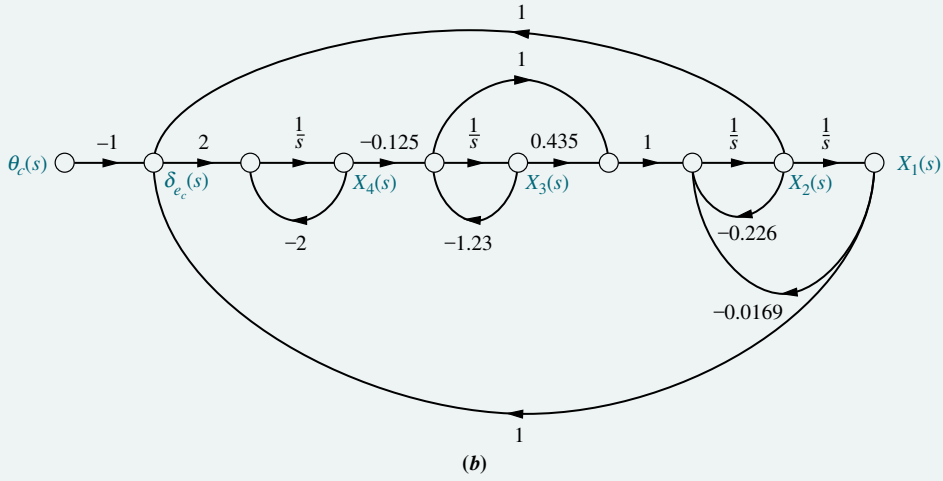
$$\dot{x}_3 = -1.23x_3 - 0.125x_4 \tag{5.115c}$$

$$\dot{x}_4 = 2x_1 + 2x_2 - 2x_4 - 2\theta c \tag{5.115d}$$

Finally, the output $y = x_1$.



(a)



(b)

FIGURE 5.37 Signal-flow graph representation of the UFSS vehicle's pitch control system: **a.** without position and rate feedback; **b.** with position and rate feedback. (Note: Explicitly required variables are $x_1 = \theta, x_2 = d\theta/dt$, and $x_4 = \delta_e$.)

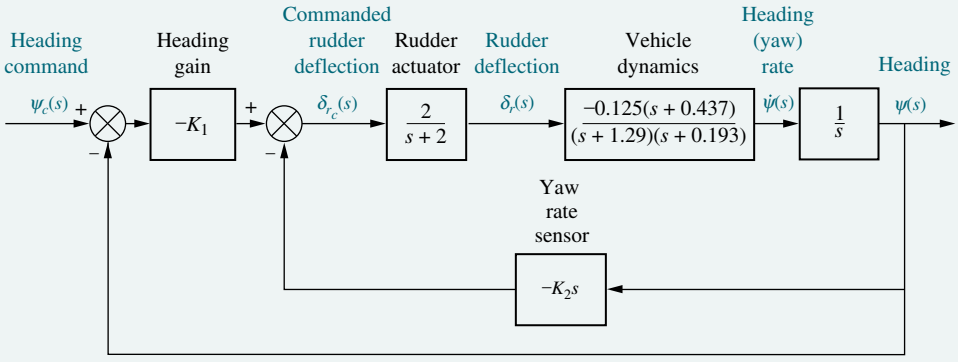


FIGURE 5.38 Block diagram of the heading control system for the UFSS vehicle

In vector-matrix form the state and output equations are

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.0169 & -0.226 & -0.795 & -0.125 \\ 0 & 0 & -1.23 & -0.125 \\ 2 & 2 & 0 & -2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} \theta_c \tag{5.116a}$$

$$y = [1 \quad 0 \quad 0 \quad 0]\mathbf{x} \tag{5.116b}$$

CHALLENGE: We now give you a problem to test your knowledge of this chapter's objectives. The UFSS vehicle steers via the heading control system shown in Figure 5.38 and repeated on the back endpapers. A heading command is the input. The input and feedback from the submersible's heading and yaw rate are used to generate a rudder command that steers the submersible (*Johnson*, *1980*). Let $K_1 = K_2 = 1$ and do the following:

a. Draw the signal-flow graph for each subsystem, making sure that heading angle, yaw rate, and rudder deflection are represented as state variables. Then interconnect the subsystems.

b. Use the signal-flow graph obtained in **a** to represent the heading control loop in state space.

MATLAB ML

c. `Use MATLAB to represent the closed-loop UFSS heading control system in state space in controller canonical form.`

Summary

One objective of this chapter has been for you to learn how to represent multiple subsystems via block diagrams or signal-flow graphs. Another objective has been to be able to reduce either the block diagram representation or the signal-flow graph representation to a single transfer function.

We saw that the block diagram of a linear, time-invariant system consisted of four elements: *signals*, *systems*, *summing junctions*, and *pickoff points*. These

elements were assembled into three basic forms: *cascade*, *parallel*, and *feedback*. Some basic operations were then derived: moving systems across summing junctions and across pickoff points.

Once we recognized the basic forms and operations, we could reduce a complicated block diagram to a single transfer function relating input to output. Then we applied the methods of Chapter 4 for analyzing and designing a second-order system for transient behavior. We saw that adjusting the gain of a feedback control system gave us partial control of the transient response.

The signal-flow representation of linear, time-invariant systems consists of two elements: nodes, which represent signals, and lines with arrows, which represent subsystems. Summing junctions and pickoff points are implicit in signal-flow graphs. These graphs are helpful in visualizing the meaning of the state variables. Also, they can be drawn first as an aid to obtaining the state equations for a system.

Mason's rule was used to derive the system's transfer function from the signal-flow graph. This formula replaced block diagram reduction techniques. Mason's rule seems complicated, but its use is simplified if there are no nontouching loops. In many of these cases, the transfer function can be written by inspection, with less labor than in the block diagram reduction technique.

Finally, we saw that systems in state space can be represented using different sets of variables. In the last three chapters, we have covered *phase-variable*, *cascade*, *parallel*, *controller canonical*, and *observer canonical* forms. A particular representation may be chosen because one set of state variables has a different physical meaning than another set, or because of the ease with which particular state equations can be solved.

In the next chapter, we discuss system stability. Without stability we cannot begin to design a system for the desired transient response. We will find out how to tell whether a system is stable and what effect parameter values have on a system's stability.

Review Questions

1. Name the four components of a block diagram for a linear, time-invariant system.
2. Name three basic forms for interconnecting subsystems.
3. For each of the forms in Question 2, state (respectively) how the equivalent transfer function is found.
4. Besides knowing the basic forms as discussed in Questions 2 and 3, what other equivalents must you know in order to perform block diagram reduction?
5. For a simple, second-order feedback control system of the type shown in Figure 5.14, describe the effect that variations of forward-path gain, K, have on the transient response.
6. For a simple, second-order feedback control system of the type shown in Figure 5.14, describe the changes in damping ratio as the gain, K, is increased over the underdamped region.
7. Name the two components of a signal-flow graph.
8. How are summing junctions shown on a signal-flow graph?
9. If a forward path touched all closed loops, what would be the value of Δ_k?
10. Name five representations of systems in state space.

State Space SS

11. Which two forms of the state-space representation are found using the same method?

State Space SS

12. Which form of the state-space representation leads to a diagonal matrix?

State Space SS

13. When the system matrix is diagonal, what quantities lie along the diagonal?

State Space SS

14. What terms lie along the diagonal for a system represented in Jordan canonical form?

State Space SS

15. What is the advantage of having a system represented in a form that has a diagonal system matrix?

State Space SS

16. Give two reasons for wanting to represent a system by alternative forms.

State Space SS

17. For what kind of system would you use the observer canonical form?

State Space SS

18. Describe state-vector transformations from the perspective of different bases.

State Space SS

19. What is the definition of an eigenvector?

State Space SS

20. Based upon your definition of an eigenvector, what is an eigenvalue?

State Space SS

21. What is the significance of using eigenvectors as basis vectors for a system transformation?

Problems

1. Reduce the block diagram shown in Figure P5.1 to a single transfer function, $T(s) = C(s)/R(s)$ Use the following methods:

   a. Block diagram reduction [Section: 5.2]

   b. MATLAB

MATLAB ML

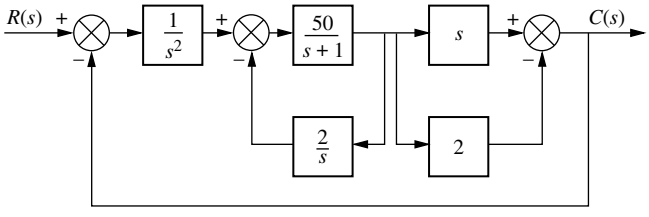


FIGURE P5.1

2. Find the closed-loop transfer function, $T(s) = C(s)/R(s)$ for the system shown in Figure P5.2, using block diagram reduction. [Section: 5.2]

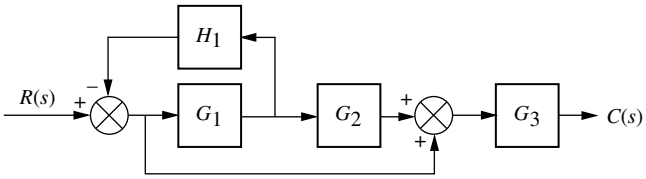


FIGURE P5.2

3. Find the equivalent transfer function, $T(s) = C(s)/R(s)$, for the system shown in Figure P5.3. [Section: 5.2]

WileyPLUS WPCS Control Solutions

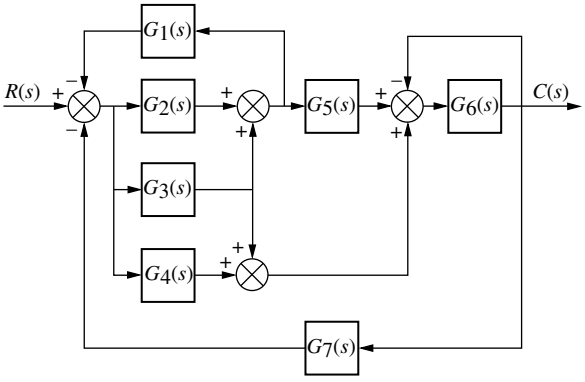


FIGURE P5.3

4. Reduce the system shown in Figure P5.4 to a single transfer function, $T(s) = C(s)/R(s)$. [Section: 5.2]

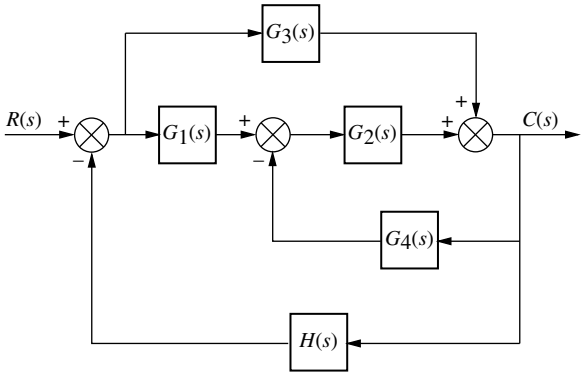


FIGURE P5.4

5. Find the transfer function, $T(s) = C(s)/R(s)$, for the system shown in Figure P5.5. Use the following methods:

a. Block diagram reduction [Section: 5.2]

b. MATLAB. Use the following transfer functions:
$G_1(s) = 1/(s+7)$, $G_2(s) = 1/(s^2+2s+3)$,
$G_3(s) = 1/(s+4)$, $G_4(s) = 1/s$,
$G_5(s) = 5/(s+7)$, $G_6(s) = 1/(s^2+5s+10)$,
$G_7(s) = 3/(s+2)$, $G_8(s) = 1/(s+6)$.
Hint: Use the **append** and **connect** commands in MATLAB's Control System Toolbox.

MATLAB ML

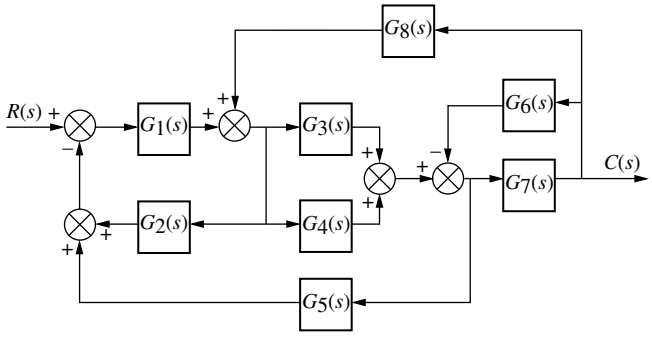


FIGURE P5.5

6. Reduce the block diagram shown in Figure P5.6 to a single block, $T(s) = C(s)/R(s)$. [Section: 5.2]

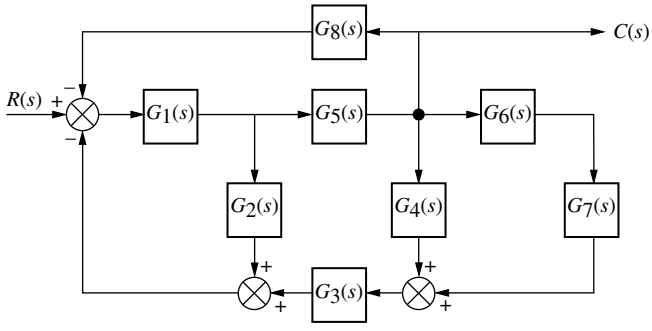


FIGURE P5.6

7. Find the unity feedback system that is equivalent to the system shown in Figure P5.7. [Section: 5.2].

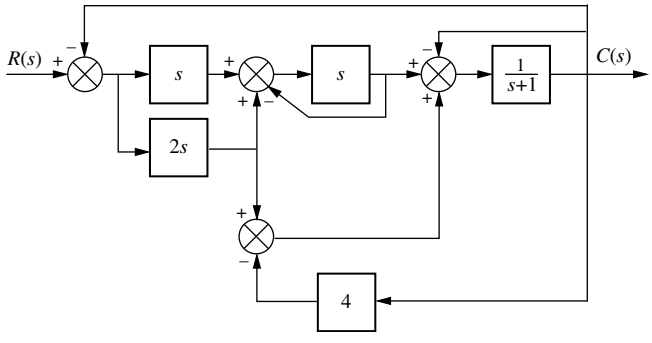


FIGURE P5.7

8. Given the block diagram of a system shown in Figure P5.8, find the transfer function $G(s) = \theta_{22}(s)/\theta_{11}(s)$. [Section: 5.2]

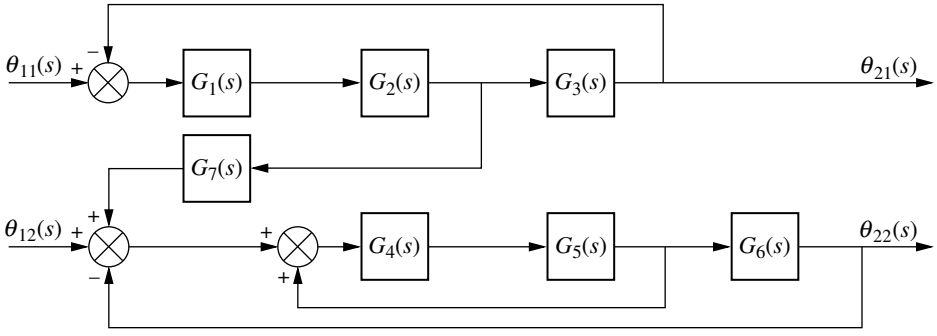


FIGURE P5.8

9. Reduce the block diagram shown in Figure P5.9 to a single transfer function, $T(s) = C(s)/R(s)$. [Section: 5.2]

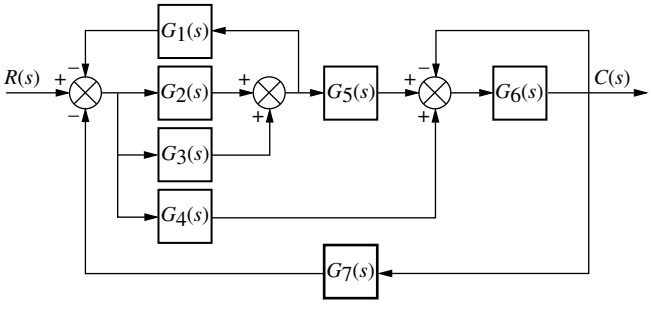


FIGURE P5.9

10. Reduce the block diagram shown in Figure P5.10 to a single block representing the transfer function, $T(s) = C(s)/R(s)$. [Section: 5.2]

WileyPLUS WPCS Control Solutions

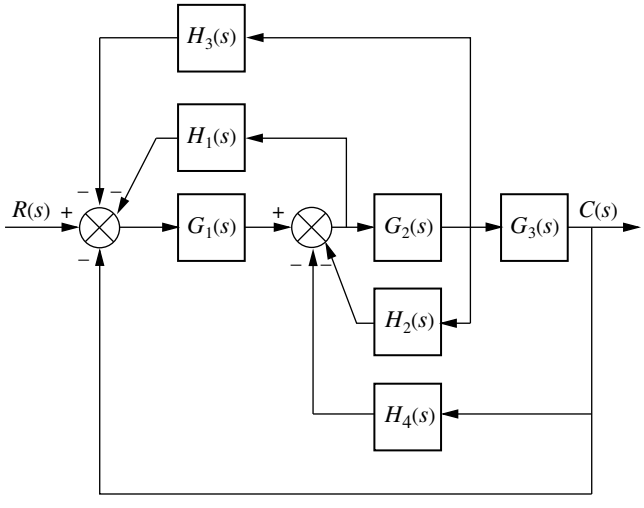


FIGURE P5.10

11. For the system shown in Figure P5.11, find the percent overshoot, settling time, and peak time for a step input if the system's response is underdamped. (Is it? Why?) [Section: 5.3]

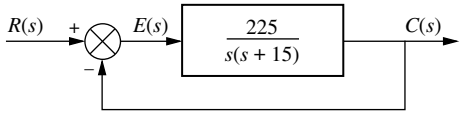


FIGURE P5.11

12. For the system shown in Figure P5.12, find the output, $c(t)$, if the input, $r(t)$, is a unit step. [Section: 5.3]

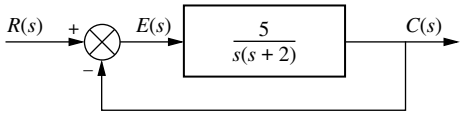


FIGURE P5.12

13. For the system shown in Figure P5.13, find the poles of the closed-loop transfer function, $T(s) = C(s)/R(s)$. [Section: 5.3]

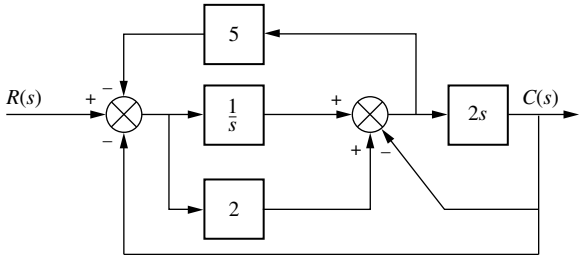


FIGURE P5.13

14. For the system of Figure P5.14, find the value of K that yields 10% overshoot for a step input. [Section: 5.3]

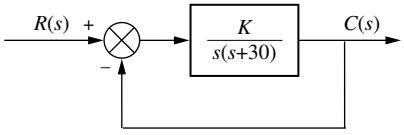


FIGURE P5.14

15. For the system shown in Figure P5.15, find K and α to yield a settling time of 0.15 second and a 30% overshoot. [Section: 5.3]

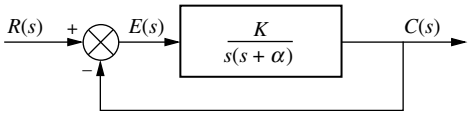


FIGURE P5.15

16. For the system of Figure P5.16, find the values of K_1 and K_2 to yield a peak time of 1.5 second and a settling time of 3.2 seconds for the closed-loop system's step response. [Section: 5.3]

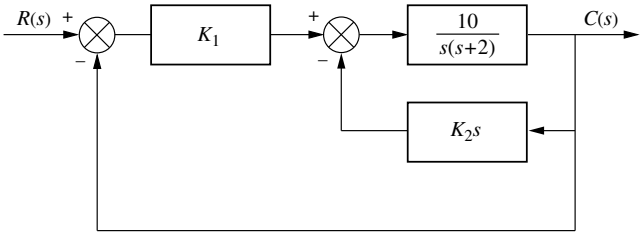


FIGURE P5.16

17. Find the following for the system shown in Figure P5.17: [Section: 5.3]

WileyPLUS WPCS Control Solutions

a. The equivalent single block that represents the transfer function, $T(s) = C(s)/R(s)$.

b. The damping ratio, natural frequency, percent overshoot, settling time, peak time, rise time, and damped frequency of oscillation.

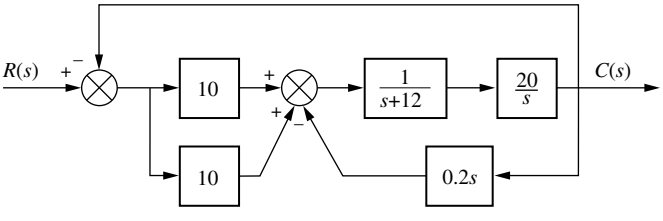


FIGURE P5.17

18. For the system shown in Figure P5.18, find ζ, ω_n, percent overshoot, peak time, rise time, and settling time. [Section: 5.3]

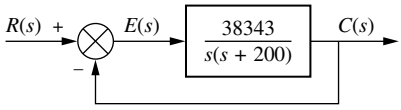


FIGURE P5.18

19. A motor and generator are set up to drive a load as shown in Figure P5.19. If the generator output voltage is $e_g(t) = K_f i_f(t)$, where $i_f(t)$ is the generator's field current, find the transfer function $G(s) = \theta_o(s)/E_i(s)$. For the generator, $K_f = 2\,\Omega$. For the motor, $K_t = 1$ N-m/A, and $K_b = 1$ V-s/rad.

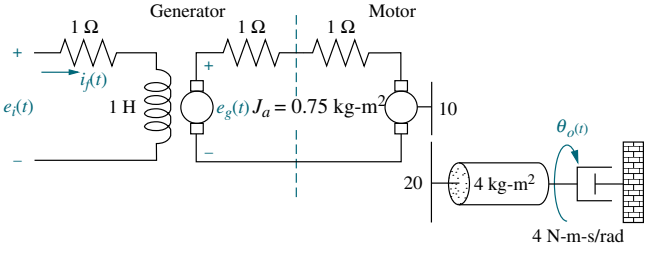


FIGURE P5.19

20. Find $G(s) = E_0(s)/T(s)$ for the system shown in Figure P5.20.

WileyPLUS WPCS Control Solutions

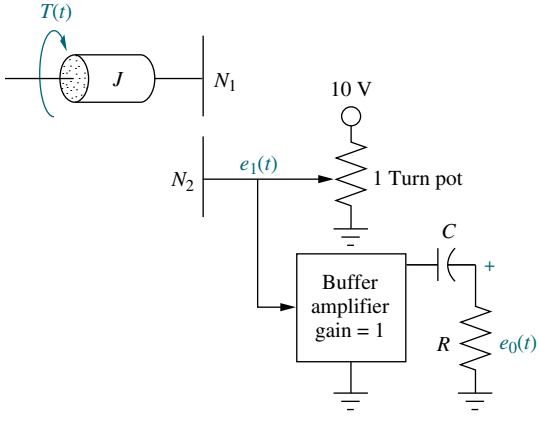


FIGURE P5.20

21. Find the transfer function $G(s) = E_o(s)/T(s)$ for the system shown in Figure P5.21.

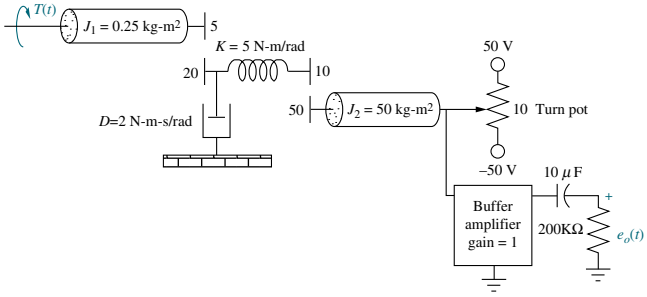


FIGURE P5.21

22. Label signals and draw a signal-flow graph for each of the block diagrams shown in the following problems: [Section: 5.4]

a. Problem 1

b. Problem 3

c. Problem 5

23. Draw a signal-flow graph for each of the following state equations: [Section: 5.6]

State Space SS

a.
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -4 & -6 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$
$$y = [1 \quad 1 \quad 0]\mathbf{x}$$

b.
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -3 & 1 \\ -3 & -4 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} r$$
$$y = [1 \quad 2 \quad 0]\mathbf{x}$$

c.
$$\dot{\mathbf{x}} = \begin{bmatrix} 7 & 1 & 0 \\ -3 & 2 & -1 \\ -1 & 0 & 2 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} r$$
$$y = [1 \quad 3 \quad 2]\mathbf{x}$$

24. Given the system below, draw a signal-flow graph and represent the system in state space in the following forms: [Section: 5.7]

State Space SS

WileyPLUS WPCS Control Solutions

a. Phase-variable form

b. Cascade form

$$G(s) = \frac{10}{(s+7)(s+8)(s+9)}$$

25. Repeat Problem 24 for

State Space SS

$$G(s) = \frac{20}{s(s-2)(s+5)(s+8)}$$

[Section: 5.7]

26. Using Mason's rule, find the transfer function, $T(s) = C(s)/R(s)$, for the system represented in Figure P5.22. [Section: 5.5]

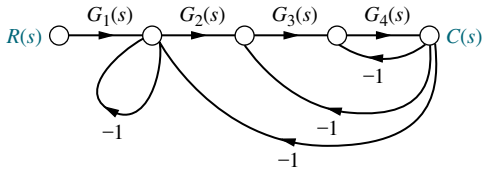


FIGURE P5.22

27. Using Mason's rule, find the transfer function, $T(s) = C(s)/R(s)$, for the system represented by Figure P5.23. [Section: 5.5]

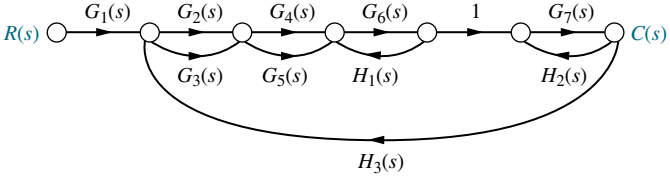


FIGURE P5.23

28. Use Mason's rule to find the transfer function of Figure 5.13 in the text. [Section: 5.5]

29. Use block diagram reduction to find the transfer function of Figure 5.21 in the text, and compare your answer with that obtained by Mason's rule. [Section: 5.5]

30. Represent the following systems in state space in Jordan canonical form. Draw the signal-flow graphs. [Section: 5.7]

State Space SS

a. $G(s) = \dfrac{(s+1)(s+2)}{(s+3)^2(s+4)}$

b. $G(s) = \dfrac{(s+2)}{(s+5)^2(s+7)^2}$

c. $G(s) = \dfrac{(s+3)}{(s+2)^2(s+4)(s+5)}$

31. Represent the systems below in state space in phase-variable form. Draw the signal-flow graphs. [Section: 5.7]

State Space SS

WileyPLUS WPCS Control Solutions

a. $G(s) = \dfrac{s+3}{s^2+2s+7}$

b. $G(s) = \dfrac{s^2 + 2s + 6}{s^3 + 5s^2 + 2s + 1}$

c. $G(s) = \dfrac{s^3 + 2s^2 + 7s + 1}{s^4 + 3s^3 + 5s^2 + 6s + 4}$

32. Repeat Problem 31 and represent each system in controller canonical and observer canonical forms. [Section: 5.7]

State Space SS

33. Represent the feedback control systems shown in Figure P5.24 in state space. When possible, represent the open-loop transfer functions separately in cascade and complete the feedback loop with the signal path from output to input. Draw your signal-flow graph to be in one-to-one correspondence to the block diagrams (as close as possible). [Section: 5.7]

State Space SS

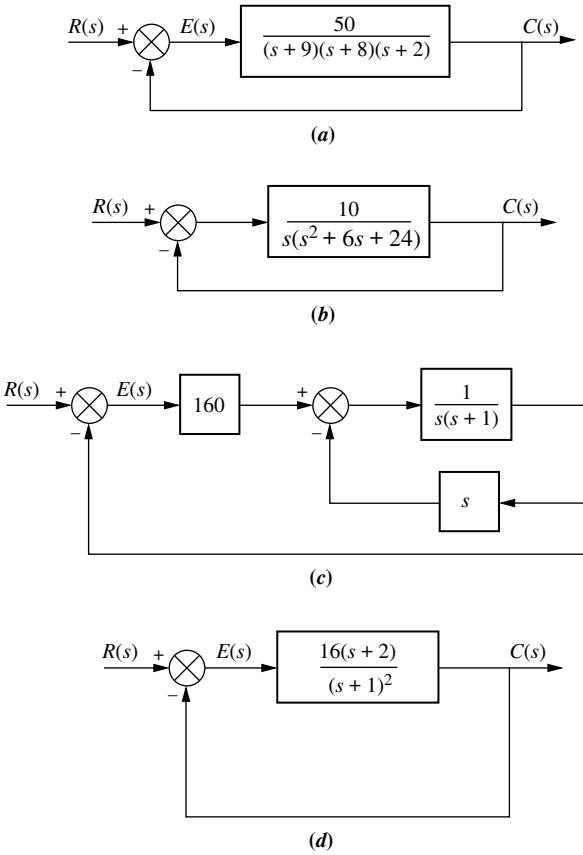


FIGURE P5.24

34. You are given the system shown in Figure P5.25. [Section: 5.7]

State Space SS

a. Represent the system in state space in phase-variable form.

b. Represent the system in state space in any other form besides phase-variable.

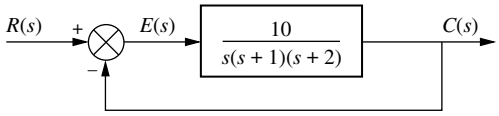


FIGURE P5.25

35. Repeat Problem 34 for the system shown in Figure P5.26. [Section: 5.7]

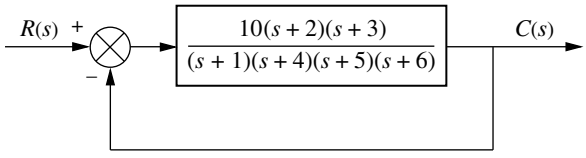


FIGURE P5.26

36. `Use MATLAB to solve Problem 35.`

MATLAB ML

37. Represent the system shown in Figure P5.27 in state space where $x_1(t)$, $x_3(t)$, and $x_4(t)$, as shown, are among the state variables, $c(t)$ is the output, and $x_2(t)$ is internal to $X_1(s)/X_3(s)$. [Section: 5.7]

State Space SS

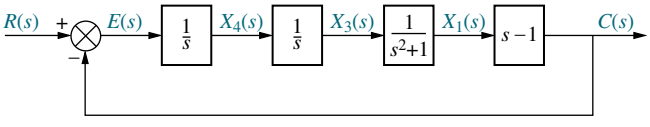


FIGURE P5.27

38. Consider the rotational mechanical system shown in Figure P5.28.

State Space SS

a. Represent the system as a signal-flow graph.

b. Represent the system in state space if the output is $\theta_2(t)$.

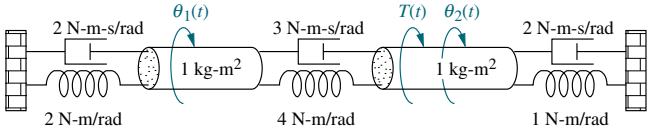


FIGURE P5.28

39. `Given a unity feedback system with the forward-path transfer function`

MATLAB ML

State Space SS

$$G(s) = \frac{7}{s(s + 9)(s + 12)}$$

use MATLAB to represent the closed loop system in state space in

a. phase-variable form;

b. parallel form.

40. Consider the cascaded subsystems shown in Figure P5.29. If $G_1(s)$ is represented in state space as

State Space SS

WileyPLUS WPCS Control Solutions

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1 r$$
$$y_1 = \mathbf{C}_1\mathbf{x}_1$$

and $G_2(s)$ is represented in state space as

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2 y_1$$
$$y_2 = \mathbf{C}_2\mathbf{x}_2$$

show that the entire system can be represented in state space as

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \cdots \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \left[\begin{array}{c:c} \mathbf{A}_1 & \mathbf{0} \\ \hdashline \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \cdots \\ \mathbf{0} \end{bmatrix} r$$

$$y_2 = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{C}_2 \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix}$$

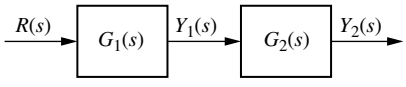


FIGURE P5.29

41. Consider the parallel subsystems shown in Figure P5.30. If $G_1(s)$ is represented in state space as

State Space SS

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1 r$$
$$y_1 = \mathbf{C}_1\mathbf{x}_1$$

and $G_2(s)$ is represented in state space as

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2 r$$
$$y_2 = \mathbf{C}_2\mathbf{x}_2$$

show that the entire system can be represented in state space as

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \cdots \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \left[\begin{array}{c:c} \mathbf{A}_1 & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{A}_2 \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \cdots \\ \mathbf{B}_2 \end{bmatrix} r$$

$$y = \left[\begin{array}{c:c} \mathbf{C}_1 & \mathbf{C}_2 \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix}$$

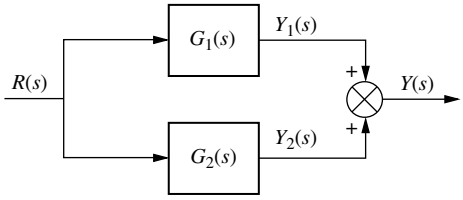


FIGURE P5.30

42. Consider the subsystems shown in Figure P5.31 and connected to form a feedback system. If $G(s)$ is represented in state space as

State Space SS

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1 e$$
$$y = \mathbf{C}_1\mathbf{x}_1$$

and $H_2(s)$ is represented in state space as

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2 y$$
$$\rho = \mathbf{C}_2\mathbf{x}_2$$

show that the closed-loop system can be represented in state space as

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \cdots \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \left[\begin{array}{c:c} \mathbf{A}_1 & -\mathbf{B}_1\mathbf{C}_2 \\ \hdashline \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1 \\ \cdots \\ \mathbf{0} \end{bmatrix} r$$

$$y = \left[\begin{array}{c:c} \mathbf{C}_1 & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix}$$

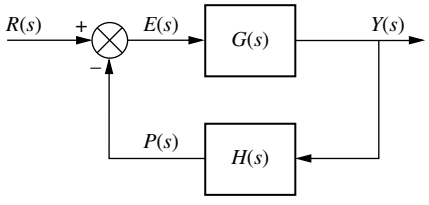


FIGURE P5.31

43. Given the system represented in state space as follows: [Section: 5.8]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -1 & -7 & 6 \\ -8 & 4 & 8 \\ 4 & 7 & -8 \end{bmatrix} \mathbf{x} + \begin{bmatrix} -5 \\ -7 \\ 5 \end{bmatrix} r$$
$$y = [-9 \quad -9 \quad -8]\mathbf{x}$$

convert the system to one where the new state vector, $\mathbf{z}$, is

$$\mathbf{z} = \begin{bmatrix} -4 & 9 & -3 \\ 0 & -4 & 7 \\ -1 & -4 & -9 \end{bmatrix} \mathbf{x}$$

44. Repeat Problem 43 for the following system: [Section: 5.8]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -5 & 1 & 1 \\ 9 & -9 & -9 \\ -9 & -1 & 8 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 9 \\ -4 \\ 0 \end{bmatrix} r$$

$$y = [-2 \quad -4 \quad 1]\mathbf{x}$$

and the following state-vector transformation:

$$\mathbf{z} = \begin{bmatrix} 5 & -4 & 9 \\ 6 & -7 & 6 \\ 6 & -5 & -3 \end{bmatrix}\mathbf{x}$$

45. Diagonalize the following system: [Section: 5.8]

State Space SS

WileyPLUS WPCS Control Solutions

$$\dot{\mathbf{x}} = \begin{bmatrix} -5 & -5 & 4 \\ 2 & 0 & -2 \\ 0 & -2 & -1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} -1 \\ 2 \\ -2 \end{bmatrix} r$$

$$y = [-1 \quad 1 \quad 2]\mathbf{x}$$

46. Repeat Problem 45 for the following system: [Section: 5.8]

State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} -10 & -3 & 7 \\ 18.25 & 6.25 & -11.75 \\ -7.25 & -2.25 & 5.75 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} r$$

$$y = [1 \quad -2 \quad 4]\mathbf{x}$$

47. Diagonalize the system in Problem 46 using MATLAB.

MATLAB ML

48. During ascent the space shuttle is steered by commands generated by the computer's guidance calculations. These commands are in the form of vehicle attitude, attitude rates, and attitude accelerations obtained through measurements made by the vehicle's inertial measuring unit, rate gyro assembly, and accelerometer assembly, respectively. The ascent digital autopilot uses the errors between the actual and commanded attitude, rates, and accelerations to gimbal the space shuttle main engines (called thrust vectoring) and the solid rocket boosters to effect the desired vehicle attitude. The space shuttle's attitude control system employs the same method in the pitch, roll, and yaw control systems. A simplified model of the pitch control system is shown in Figure P5.32.[4]

a. Find the closed-loop transfer function relating actual pitch to commanded pitch. Assume all other inputs are zero.

b. Find the closed-loop transfer function relating actual pitch rate to commanded pitch rate. Assume all other inputs are zero.

c. Find the closed-loop transfer function relating actual pitch acceleration to commanded pitch acceleration. Assume all other inputs are zero.

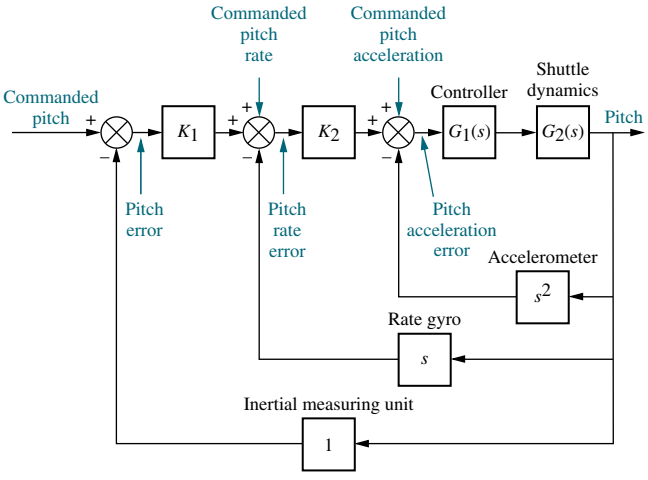


FIGURE P5.32 Space shuttle pitch control system (simplified)

49. An AM radio modulator generates the product of a carrier waveform and a message waveform, as shown in Figure P5.33 (*Kurland, 1971*). Represent the system in state space if the carrier is a sinusoid of frequency $\omega = a$, and the message is a sinusoid of frequency $\omega = b$. Note that this system is nonlinear because of the multiplier.

State Space SS

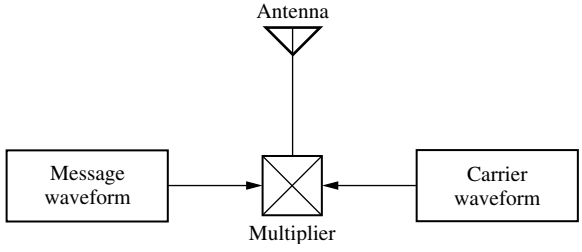


FIGURE P5.33 AM modulator

50. A model for human eye movement consists of the closed-loop system shown in Figure P5.34, where an

[4] Source of background information for this problem: Rockwell International.

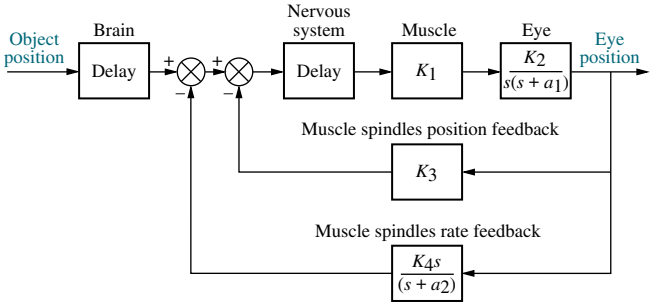


FIGURE P5.34 Feedback control system representing human eye movement

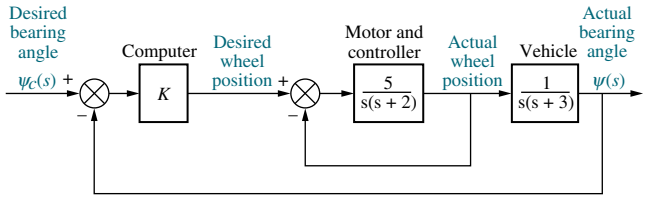
object's position is the input and the eye position is the output. The brain sends signals to the muscles that move the eye. These signals consist of the difference between the object's position and the position and rate information from the eye sent by the muscle spindles. The eye motion is modeled as an inertia and viscous damping and assumes no elasticity (spring) (*Milhorn*, *1966*). Assuming that the delays in the brain and nervous system are negligible, find the closed-loop transfer function for the eye position control.

51. A HelpMate transport robot, shown in Figure P5.35(*a*), is used to deliver goods in a hospital setting. The robot can deliver food, drugs, laboratory materials, and patients' records (*Evans*, *1992*). Given the simplified block diagram of the robot's bearing angle control system, as shown in Figure P5.35(*b*), do the following:

WileyPLUS WPCS Control Solutions



(*a*)



(*b*)

FIGURE P5.35 **a**. HelpMate robot used for in-hospital deliveries; **b**. simplified block diagram for bearing angle control

a. Find the closed-loop transfer function.

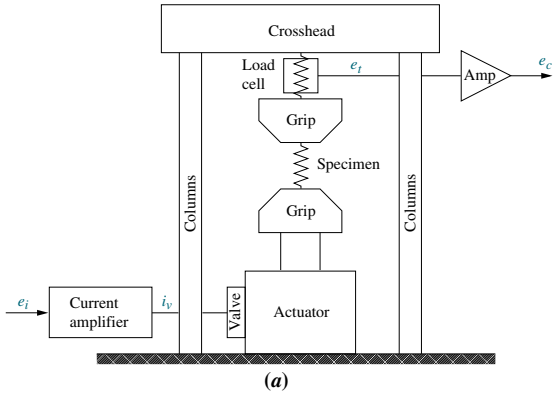
b. Represent the system in state space, where the input is the desired bearing angle, the output is the actual bearing angle, and the actual wheel position and actual bearing angle are among the state variables.

State Space SS

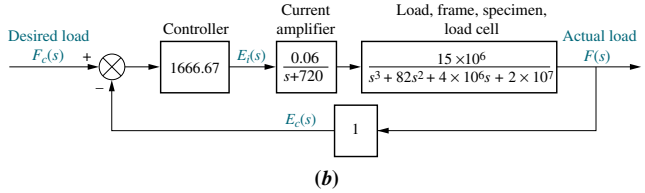
c. Simulate the closed-loop system using MATLAB. Obtain the unit step response for different values of K that yield responses from overdamped to underdamped to unstable.

MATLAB ML

52. Automatically controlled load testers can be used to test product reliability under real-life conditions. The tester consists of a load frame and specimen as shown in Figure P5.36(*a*). The desired load is



(*a*)



(*b*)

FIGURE P5.36 **a**. Load tester, (© 1992 IEEE) **b.** approximate block diagram

input via a voltage, $e_i(t)$, to a current amplifier. The output load is measured via a voltage, $e_i(t)$, from a load cell measuring the load on the specimen. Figure P5.36(b) shows an approximate model of a load testing system without compensation (*Bailey*, *1992*).

a. Model the system in state space. State Space SS

b. Simulate the step response using MATLAB. Is the response predominantly first or second order? Describe the characteristics of the response that need correction. MATLAB ML

53. Consider the F4-E aircraft of Problem 22, Chapter 3. If the open-loop transfer function relating normal acceleration, $A_n(s)$, to the input deflection command, $\delta_c(s)$, is approximated as State Space SS

$$\frac{A_n(s)}{\delta_c(s)} = \frac{-272(s^2 + 1.9s + 84)}{(s + 14)(s - 1.8)(s + 4.9)}$$

(*Cavallo*, *1992*), find the state-space representation in

a. Phase-variable form

b. Controller canonical form

c. Observer canonical form

d. Cascade form

e. Parallel form

54. Find the closed-loop transfer function of the Unmanned Free-Swimming Submersible vehicle's pitch control system shown on the back endpapers (*Johnson*, *1980*).

55. Repeat Problem 54 using MATLAB. MATLAB ML

56. Use Simulink to plot the effects of nonlinearities upon the closed-loop step response of the antenna azimuth position control system shown on the front endpapers, Configuration 1. In particular, consider individually each of the following nonlinearities: saturation (±5 volts), backlash (dead-band width 0.15), dead-zone (−2 to +2), as well as the linear response. Assume the preamplifier gain is 100 and the step input is 2 radians. Simulink SL

57. Problem 12 in Chapter 1 describes a high-speed proportional solenoid valve. A subsystem of the valve is the solenoid coil shown in Figure P5.37. Current through the coil, L, generates a magnetic field that produces a force to operate the valve. Figure P5.37can be represented as a block diagram (*Vaughan*, *1996*)

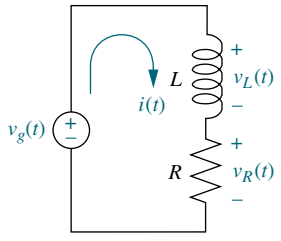


FIGURE P5.37 Solenoid coil circuit

a. Derive a block diagram of a feedback system that represents the coil circuit, where the applied voltage, $v_g(t)$, is the input, the coil voltage, $v_L(t)$, is the error voltage, and the current, $i(t)$, is the output.

b. For the block diagram found in Part **a**, find the Laplace transform of the output current, $I(s)$.

c. Solve the circuit of Figure P5.37 for $I(s)$, and compare to your result in Part **b**.

58. Ktesibios' water clock (see Section 1.2) is probably the first man-made system in which feedback was used in a deliberate manner. Its operations are shown in Figure P5.38(a). The clock indicates time progressively on scale D as water falls from orifice A toward vessel B. Clock accuracy depends mainly on water height h_f in the water reservoir G, which must be maintained at a constant level h_r by means of the conical float F that moves up or down to control the water inflow. Figure P5.38(b) shows a block diagram describing the system (*Lepschy*, *1992*).

Let $q_i(t)$ and $q_o(t)$ represent the input and output water flow, respectively, and h_m the height of water in vessel B. Use Mason's rule to find the following transfer functions, assuming α and β are constants:

a. $\dfrac{H_m(s)}{H_r(s)}$

b. $\dfrac{H_f(s)}{H_r(s)}$

c. $\dfrac{Q_i(s)}{H_r(s)}$

d. $\dfrac{Q_o(s)}{H_r(s)}$

e. Using the above transfer functions, show that if $h_r(t) =$ constant, then $q_o(t) =$ constant and $h_m(t)$ increases at a constant speed.

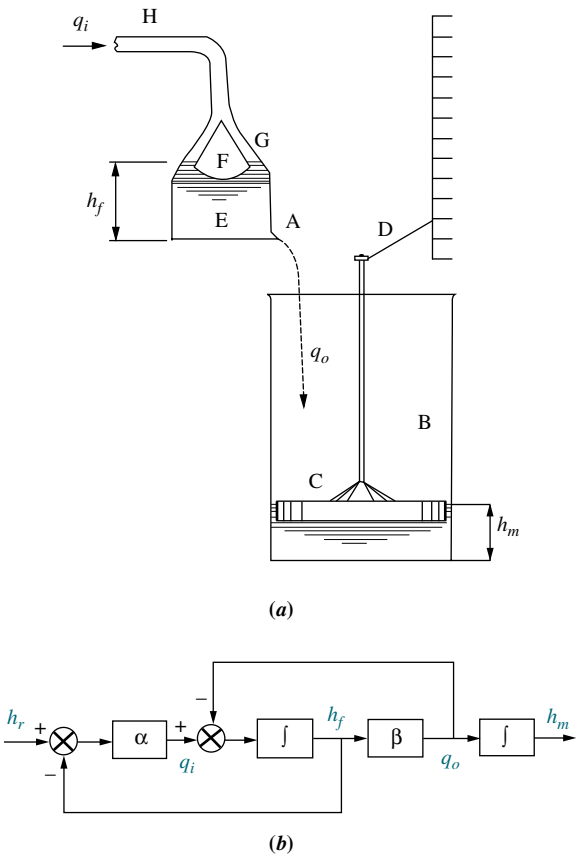


FIGURE P5.38 **a.** Ktesibios' water clock; **b.** water clock block diagram (© 1992 IEEE)

59. Some robotic applications can benefit from actuators in which load position as well as exerted force are controlled. Figure P5.39 shows the block diagram of such an actuator, where u_1 and u_2 are voltage inputs to two coils, each of which controls a pneumatic piston, and y represents the load displacement.

The system's output is u, the differential pressure acting on the load. The system also has a disturbance input f_{ext}, which represents external forces that are not system generated, but are acting on the load. A is a constant (*Ben-Dov, 1995*). Use any method to obtain:

a. An expression for the system's output in terms of the inputs u_1 and u_2 (Assume $f_{\text{ext}} = 0$.)

b. An expression for the effect of f_{ext} on the output u (Assume u_1 and $u_2 = 0$.)

c. What condition on the inputs u_1 and u_2 will result in $u = 0$?

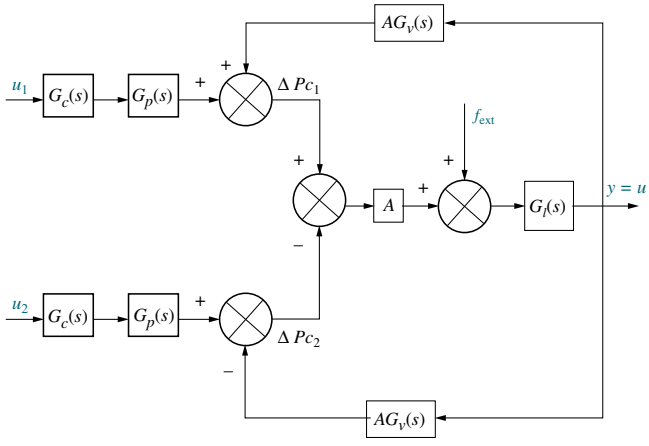


FIGURE P5.39 Actuator block diagram (© 1995 IEEE)

60. Figure P5.40 shows a noninverting operational amplifier.

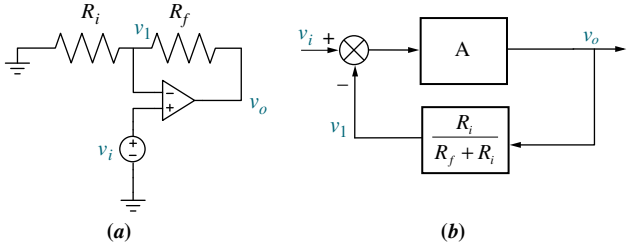


FIGURE P5.40 **a.** Noninverting amplifier; **b.** block diagram

Assuming the operational amplifier is ideal,

a. Verify that the system can be described by the following two equations:

$$v_o = A(v_i - v_o)$$
$$v_1 = \frac{R_i}{R_i + R_f} v_o$$

b. Check that these equations can be described by the block diagram of Figure P5.40(b)

c. Use Mason's rule to obtain the closed-loop system transfer function $\dfrac{V_o(s)}{V_i(s)}$

d. Show that when $A \to \infty$, $\dfrac{V_o(s)}{V_i(s)} = 1 + \dfrac{R_f}{R_i}$

61. Figure P5.41 shows the diagram of an inverting operational amplifier.

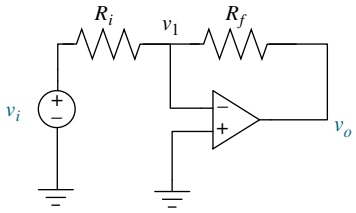


FIGURE P5.41 Inverting operational amplifier

a. Assuming an ideal operational amplifier, use a similar procedure to the one outlined in Problem 60 to find the system equations.

b. Draw a corresponding block diagram and obtain the transfer function $\dfrac{V_o(s)}{V_i(s)}$.

c. Show that when $A \to \infty$, $\dfrac{V_o(s)}{V_i(s)} = -\dfrac{R_f}{R_i}$.

62. Figure P5.42(*a*) shows an *n*-channel enhancement-mode MOSFET source follower circuit. Figure P5.42(*b*) shows its small-signal equivalent (where $R_i = R_1 \| R_2$) (*Neamen, 2001*).

a. Verify that the equations governing this circuit are

$$\frac{v_{in}}{v_i} = \frac{R_i}{R_i + R_s}; \quad v_{gs} = v_{in} - v_o; \quad v_o = g_m(R_s \| r_o)v_{gs}$$

b. Draw a block diagram showing the relations between the equations.

c. Use the block diagram in Part **b** to find $\dfrac{V_o(s)}{V_i(s)}$.

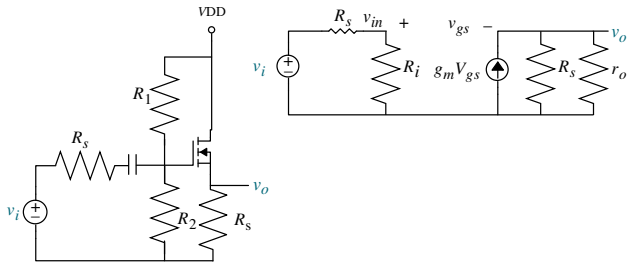


FIGURE P5.42 **a.** An *n*-channel enhancement-mode MOSFET source follower circuit; **b.** small-signal equivalent

63. A car active suspension system adds an active hydraulic actuator in parallel with the passive damper and spring to create a dynamic impedance that responds to road variations. The block diagram of Figure P5.43 depicts such an actuator with closed-loop control.

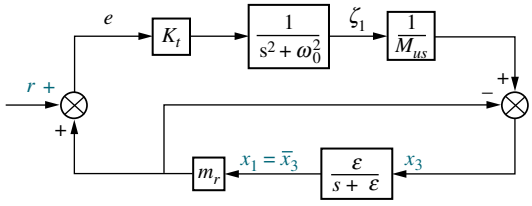


FIGURE P5.43 (© 1997 IEEE)

In the figure, K_t is the spring constant of the tire, M_{US} is the wheel mass, r is the road disturbance, x_1 is the vertical car displacement, x_3 is the wheel vertical displacement, $\omega_0^2 = \frac{K_t}{M_{US}}$ is the natural frequency of the unsprung system and ε is a filtering parameter to be judiciously chosen (*Lin, 1997*). Find the two transfer functions of interest:

a. $\dfrac{X_3(s)}{R(s)}$

b. $\dfrac{X_1(s)}{R(s)}$

64. The basic unit of skeletal and cardiac muscle cells is a *sarcomere*, which is what gives such cells a striated (parallel line) appearance. For example, one bicep cell has about 10^5 sarcomeres. In turn, sarcomeres are composed of protein complexes. Feedback mechanisms play an important role in sarcomeres and thus muscle contraction. Namely, Fenn's law says that the energy liberated during muscle contraction depends on the initial conditions and the load encountered. The following linearized model describing sarcomere contraction has been developed for cardiac muscle:

$$\begin{bmatrix} \dot{A} \\ \dot{T} \\ \dot{U} \\ \dot{SL} \end{bmatrix} = \begin{bmatrix} -100.2 & -20.7 & -30.7 & 200.3 \\ 40 & -20.22 & 49.95 & 526.1 \\ 0 & 10.22 & -59.95 & -526.1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A \\ T \\ U \\ SL \end{bmatrix} + \begin{bmatrix} 208 \\ -208 \\ -108.8 \\ -1 \end{bmatrix} u(t)$$

$$y = [\,0 \quad 1570 \quad 1570 \quad 59400\,] \begin{bmatrix} A \\ T \\ U \\ SL \end{bmatrix} - 6240u(t)$$

where

A = density of regulatory units with bound calcium and adjacent weak cross bridges (μM)
T = density of regulatory units with bound calcium and adjacent strong cross bridges (M)
U = density of regulatory units without bound calcium and adjacent strong cross bridges (M)
SL = sarcomere length (m)

The system's input is $u(t)$ = the shortening muscle velocity in meters/second and the output is $y_{(t)}$ = muscle force output in Newtons (*Yaniv, 2006*).

Do the following:

a. Use MATLAB to obtain the transfer function $\frac{Y(s)}{U(s)}$. MATLAB ML

b. Use MATLAB to obtain a partial-fraction expansion for $\frac{Y(s)}{U(s)}$. MATLAB ML

c. Draw a signal-flow diagram of the system in parallel form. State Space SS

d. Use the diagram of Part **c** to express the system in state-variable form with decoupled equations. State Space SS

65. An electric ventricular assist device (EVAD) has been designed to help patients with diminished but still functional heart pumping action to work in parallel with the natural heart. The device consists of a brushless dc electric motor that actuates on a pusher plate. The plate movements help the ejection of blood in systole and sac filling in diastole. System dynamics during systolic mode have been found to be:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{P}_{ao} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -68.3 & -7.2 \\ 0 & 3.2 & -0.7 \end{bmatrix} \begin{bmatrix} x \\ v \\ P_{ao} \end{bmatrix} + \begin{bmatrix} 0 \\ 425.4 \\ 0 \end{bmatrix} e_m$$

The state variables in this model are x, the pusher plate position, v, the pusher plate velocity, and P_{ao}, the aortic blood pressure. The input to the system is e_m, the motor voltage (*Tasch, 1990*).

a. Use MATLAB to find a similarity transformation to diagonalize the system. MATLAB ML

b. Use MATLAB and the obtained similarity transformation of Part a to obtain a diagonalized expression for the system. MATLAB ML

66. In an experiment to measure and identify postural arm reflexes, subjects hold with their hands a linear hydraulic manipulator. A load cell is attached to the actuator handle to measure resulting forces. At the application of a force, subjects try to maintain a fixed posture. Figure P5.44 shows a block diagram for the combined arm-environment system.

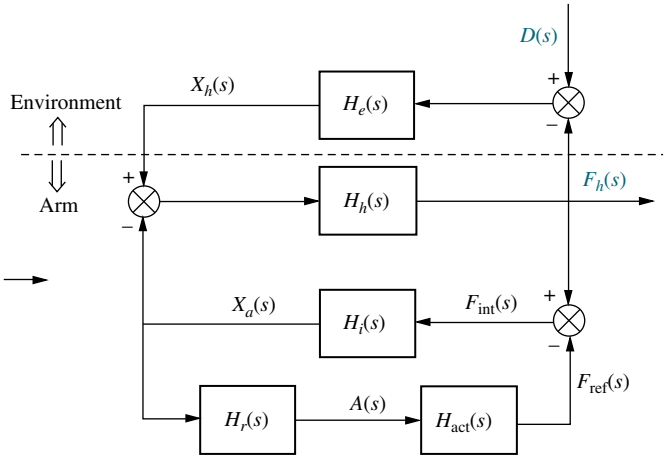


FIGURE P5.44

In the diagram, $H_r(s)$ represents the reflexive length and velocity feedback dynamics; $H_{\text{act}}(s)$ the activation dynamics, $H_i(s)$ the intrinsic act dynamics; $H_h(s)$ the hand dynamics; $H_e(s)$ the environmental dynamics; $X_a(s)$ the position of the arm; $X_h(s)$ the measured position of the hand; $F_h(s)$ the measured interaction force applied by the hand; $F_{\text{int}}(s)$ the intrinsic force; $F_{\text{ref}}(s)$ the reflexive force; $A(s)$ the reflexive activation; and $D(s)$ the external force perturbation (*de Vlugt, 2002*).

a. Obtain a signal-flow diagram from the block diagram.

b. Find $\frac{F_h(s)}{D(s)}$.

67. Use LabVIEW's Control Design and Simulation Module to obtain the controller and the observer canonical forms for: State Space SS LabVIEW LV

$$G(s) = \frac{s^2 + 7s + 2}{s^3 + 9s^2 + 26s + 24}$$

68. A virtual reality simulator with haptic (sense of touch) feedback was developed to simulate the control of a submarine driven through a joystick input. Operator haptic feedback is provided through joystick position constraints and simulator movement (*Karkoub, 2010*). Figure P5.45 shows the block diagram of the haptic feedback system in which the input u_h is the force exerted by the muscle of the human arm; and the outputs are y_s, the position of the simulator, and y_j, the position of the joystick.

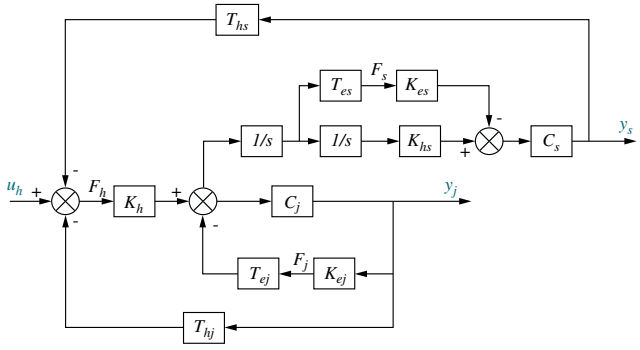


FIGURE P5.45 Copyright © 2010 Cambridge University Press. Reprinted with permission.

a. Find the transfer function $\dfrac{Y_s(s)}{U_h(s)}$.

b. Find the transfer function $\dfrac{Y_j(s)}{U_h(s)}$.

69. Some medical procedures require the insertion of a needle under a patient's skin using CT scan monitoring guidance for precision. CT scans emit radiation, posing some cumulative risks for medical personnel. To avoid this problem, a remote control robot has been developed (*Piccin, 2009*). The robot controls the needle in position and angle in the constraint space of a CT scan machine and also provides the physician with force feedback commensurate with the insertion opposition encountered by the type of tissue in which the needle is inserted. The robot has other features that give the operator the similar sensations and maneuverability as if the needle was inserted directly. Figure P5.46 shows the block diagram of the force insertion mechanism, where F_h is the input force and X_h is the output displacement. Summing junction inputs are positive unless indicated with a negative sign. By way of explanation, Z = impedance; G = transfer function; C_i = communication channel transfer functions; F = force; X = position. Subscripts h and m refer to the master manipulator. Subscripts s and e refer to the slave manipulator.

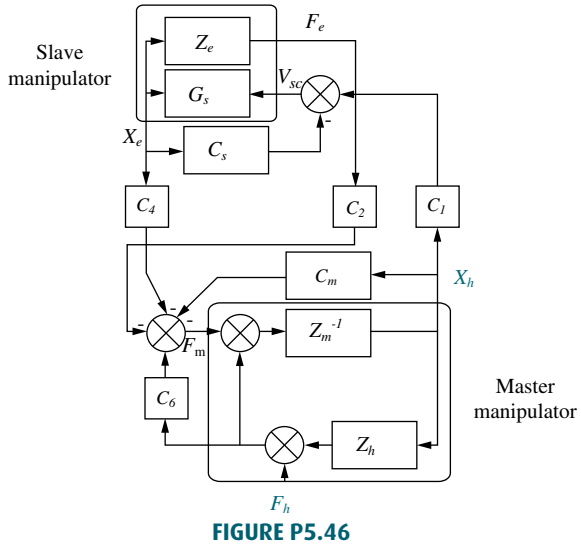


FIGURE P5.46

a. Assuming $Z_h = 0$, $C_1 = C_s$, $C_2 = 1 + C_6$ and $C_4 = -C_m$ use Mason's Rule to show that the transfer function from the operators force input F_h to needle displacement X_h is given by

$$Y(s) = \frac{X_h(s)}{F_h(s)} = \frac{Z_m^{-1}C_2(1 + G_sC_s)}{1 + G_sC_s + Z_m^{-1}(c_m + C_2Z_eG_sC_s)}$$

b. Now with $Z_h \neq 0$ show that $\dfrac{X_h(s)}{F_h(s)} = \dfrac{Y(s)}{1 + Y(s)Z_h}$

70. A hybrid solar cell and diesel power distribution system has been proposed and tested (*Lee, 2007*). The system has been shown to have a very good uninterruptible power supply as well as line voltage regulation capabilities. Figure P5.47 shows a signal-flow diagram of the system. The output, V_{Load}, is the voltage across the load. The two inputs are I_{Cf}, the reference current, and I_{Dist}, the disturbance representing current changes in the supply.

a. Refer to Figure P5.47 and find the transfer function $\dfrac{V_{Load}(s)}{I_{Cf}(s)}$.

b. Find the transfer function $\dfrac{V_{Load}(s)}{I_{Dist}(s)}$.

71. Continuous casting in steel production is essentially a solidification process by which molten steel is

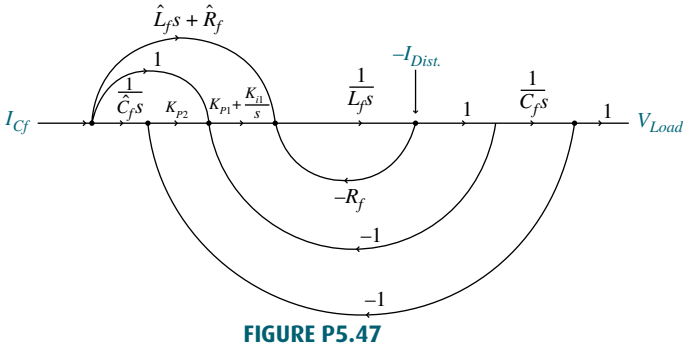


FIGURE P5.47

solidified into a steel slab after passing through a mold, as shown in Figure P5.48(*a*). Final product dimensions depend mainly on the casting speed V_p (in m/min), and on the stopper position X(in %) that controls the flow of molten material into the mold (*Kong, 1993*). A simplified model of a casting system is shown in Figure P5.48(*b*) (*Kong, 1993*) and (*Graebe, 1995*). In the model, H_m = mold level (in mm); H_t = assumed constant height of molten steel in the tundish; D_z = mold thickness = depth of nozzle immerged into molten steel; and W_t = weight of molten steel in the tundish.

For a specific setting let $A_m = 0.5$ and

$$G_x(s) = \frac{0.63}{s + 0.926}$$

Also assume that the valve positioning loop may be modeled by the following second-order transfer function:

$$G_V(s) = \frac{X(s)}{Y_C(s)} = \frac{100}{s^2 + 10s + 100}$$

and the controller is modeled by the following transfer function:

$$G_C(s) = \frac{1.6(s^2 + 1.25s + 0.25)}{s}$$

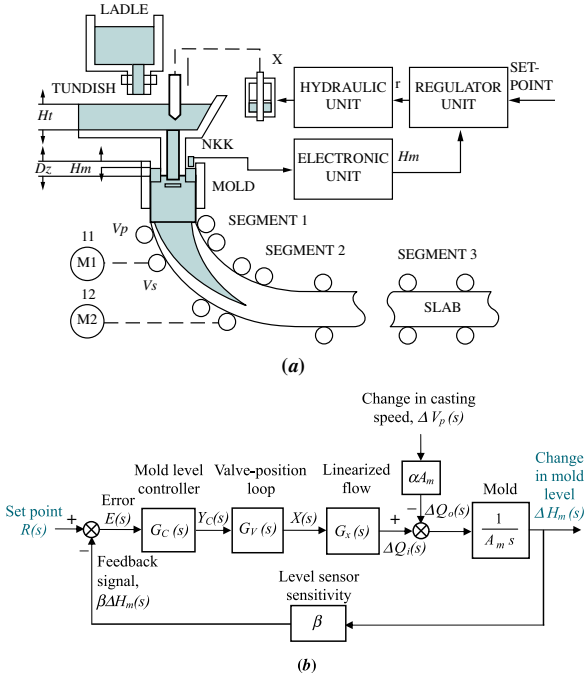


FIGURE P5.48 Steel mold process: **a.** process (© 1993 IEEE); **b.** block diagram

The sensitivity of the mold level sensor is $\beta = 0.5$ and the initial values of the system variables at $t = 0^-$ are: $R(0^-) = 0$; $Y_C(0^-) = X(0^-) = 41.2$; $\Delta H_m(0^-) = 0$; $H_m(0^-) = -75$; $\Delta V_p(0^-) = 0$; and $V_p(0^-) = 0$. Do the following:

a. Assuming $v_p(t)$ is constant $[\Delta v_p = 0]$, find the closed-loop transfer function $T(s) = \Delta H_m(s)/R(s)$.

Simulink SL

b. For $r(t) = 5\,u(t)$, $v_p(t) = 0.97\,u(t)$, and $H_m(0^-) = -75$ mm, use Simulink to simulate the system. Record the time and mold level (in array format) by connecting them to **Workspace** sinks, each of which should carry the respective variable name. After the simulation ends, utilize MATLAB plot commands to obtain and edit the graph of $h_m(t)$ from $t = 0$ to 80 seconds.

72. A simplified second-order transfer function model for bicycle dynamics is given by

State Space SS

$$\frac{\varphi(s)}{\delta(s)} = \frac{aV}{bh} \frac{\left(s + \frac{V}{a}\right)}{\left(s^2 - \frac{g}{h}\right)}$$

The input is $\delta(s)$, the steering angle, and the output is $\varphi(s)$, the tilt angle (between the floor and the bicycle longitudinal plane). In the model parameter a is the horizontal distance from the center of the back wheel to the bicycle center of mass; b is the horizontal distance between the centers of both wheels; h is the vertical distance from the center of mass to the floor; V is the rear wheel velocity (assumed constant); and g is the gravity constant. It is also assumed that the rider remains at a fixed position with respect to the bicycle so that the steer axis is vertical and that all angle deviations are small (*Åstrom, 2005*).

a. Obtain a state-space representation for the bicycle model in phase-variable form.

b. Find system eigenvalues and eigenvectors.

c. Find an appropriate similarity transformation matrix to diagonalize the system and obtain the state-space system's diagonal representation.

73. It is shown in Figure 5.6(*c*) that when negative feedback is used, the overall transfer function for the system of Figure 5.6(*b*) is

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Develop the block diagram of an alternative feedback system that will result in the same closed-loop transfer function, $C(s)/R(s)$, with $G(s)$ unchanged and unmoved. In addition, your new block diagram must have unity gain in the feedback path. You can add input transducers and/or controllers in the main forward path as required.

DESIGN PROBLEMS

74. The motor and load shown in Figure P5.49(a) are used as part of the unity feedback system shown in Figure P5.49(b). Find the value of the coefficient of viscous damping, D_L, that must be used in order to yield a closed-loop transient response having a 20% overshoot.

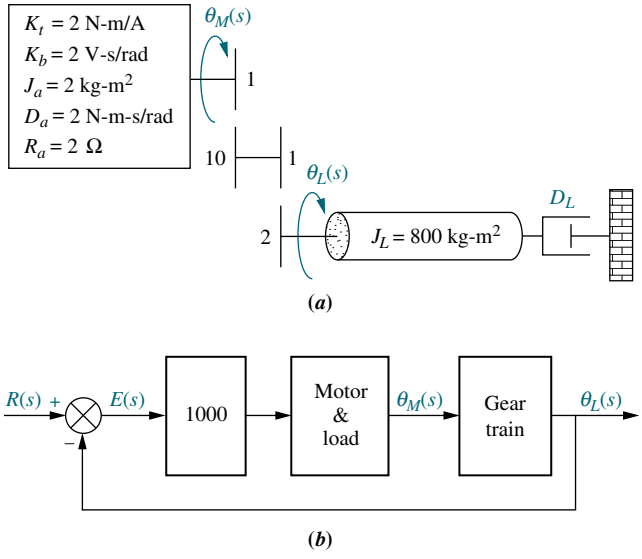


FIGURE P5.49 Position control: **a.** motor and load; **b.** block diagram

75. Assume that the motor whose transfer function is shown in Figure P5.50(a) is used as the forward path of a closed-loop, unity feedback system.

a. Calculate the percent overshoot and settling time that could be expected.

b. You want to improve the response found in Part **a**. Since the motor and the motor constants cannot be changed, an amplifier and a tachometer (voltage generator) are inserted into the loop, as shown in Figure P5.50. Find the values of K_1 and K_2 to yield a 16% overshoot and a settling time of 0.2 second.

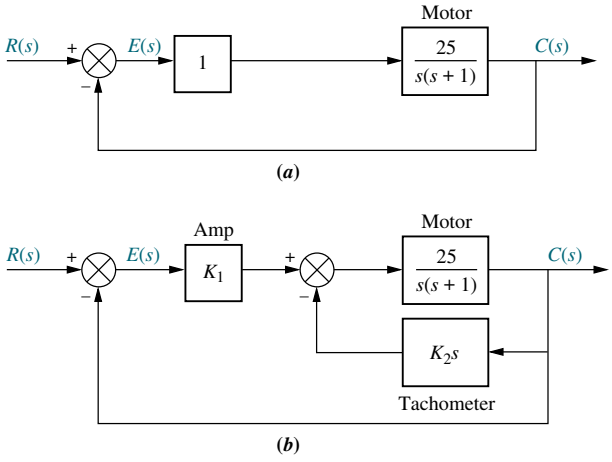


FIGURE P5.50 **a.** Position control; **b.** position control with tachometer

76. The system shown in Figure P5.51 will have its transient response altered by adding a tachometer. Design K and K_2 in the system to yield a damping ratio of 0.69. The natural frequency of the system before the addition of the tachometer is 10 rad/s.

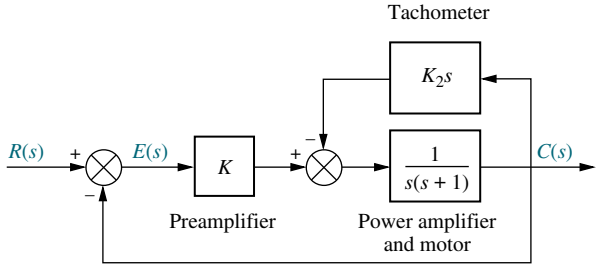


FIGURE P5.51 Position control

77. The mechanical system shown in Figure P5.52(a) is used as part of the unity feedback system shown in Figure P5.52(b). Find the values of M and D to yield 20% overshoot and 2 seconds settling time.

WileyPLUS WPCS Control Solutions

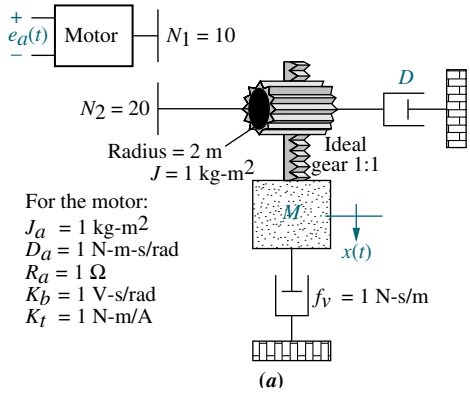
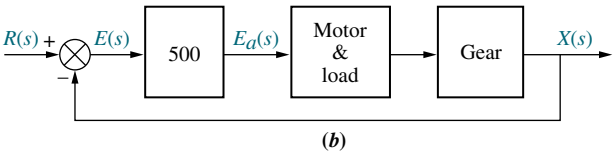


FIGURE P5.52 **a.** Motor and load; (*figure continues*)



(*b*)

FIGURE P5.52 (*Continued*) **b.** motor and load in feedback system

78. Assume ideal operational amplifiers in the circuit of Figure P5.53.

a. Show that the leftmost operational amplifier works as a subtracting amplifier. Namely, $v_1 = v_o - v_{in}$.

b. Draw a block diagram of the system, with the subtracting amplifier represented with a summing junction, and the circuit of the rightmost operational amplifier with a transfer function in the forward path. Keep R as a variable.

c. Obtain the system's closed-loop transfer function.

d. For a unit step input, obtain the value of R that will result in a settling time $T_s = 1$ msec.

e. Using the value of R calculated in Part **d**, make a sketch of the resulting unit step response.

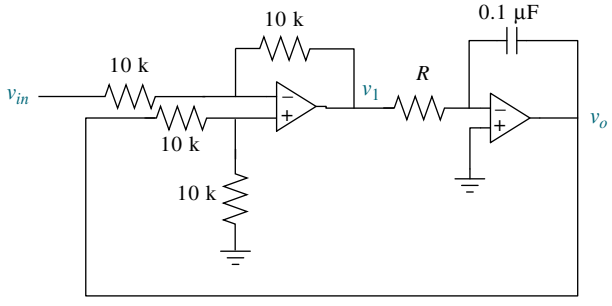


FIGURE P5.53

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

79. High-speed rail pantograph. Problem 21 in Chapter 1 discusses the active control of a pantograph mechanism for high-speed rail systems. In this problem you found a functional block diagram relating the output force (*actual*) to the input force (*desired* output). In Problem 67, Chapter 2, you found the transfer function for the pantograph dynamics, that is, the transfer function relating the displacement of the spring that models the head to the applied force, or $G(s) = (Y_h(s) - Y_{cat}(s))/F_{up}(s)$ (*O'Connor, 1997*). We now create a pantograph active-control loop by adding the following components and following your functional block diagram found in Problem 21, Chapter 1: input transducer $(G_i(s) = 1/100)$, controller $(G_c(s) = K)$, actuator $(G_a(s) = 1/1000)$, pantograph spring $(K_s = 82.3 \times 10^3 \text{N/m})$, and sensor $(H_o(s) = 1/100)$.

a. Using the functional block diagram from your solution of Problem 21 in Chapter 1, and the pantograph dynamics, $G(s)$, found in Problem 67, Chapter 2, assemble a block diagram of the active pantograph control system.

b. Find the closed-loop transfer function for the block diagram found in Part **a** if $K = 1000$.

c. Represent the pantograph dynamics in phase-variable form and find a state-space representation for the closed-loop system if $K = 1000$). State Space SS

80. Control of HIV/AIDS. Given the HIV system of Problem 82 in Chapter 4 and repeated here for convenience (*Craig, 2004*): State Space SS

$$\begin{bmatrix} \dot{T} \\ \dot{T}^* \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -0.04167 & 0 & -0.0058 \\ 0.0217 & -0.24 & 0.0058 \\ 0 & 100 & -2.4 \end{bmatrix} \begin{bmatrix} T \\ T^* \\ v \end{bmatrix} + \begin{bmatrix} 5.2 \\ -5.2 \\ 0 \end{bmatrix} u_1$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T \\ T^* \\ v \end{bmatrix}$$

Express the system in the following forms:

a. Phase-variable form

b. Controller canonical form

c. Observer canonical form

Finally,

d. `Use MATLAB to obtain the system's diagonalized representation.` MATLAB ML

81. Hybrid vehicle. Figure P5.54 shows the block diagram of a possible cascade control scheme for an HEV driven by a dc motor (*Preitl, 2007*).

Let the speed controller $G_{SC}(s) = 100 + \frac{40}{s}$, the torque controller and power amp $K_A G_{TC}(s) = 10 + \frac{6}{s}$, the current sensor sensitivity $K_{CS} = 0.5$, the speed sensor sensitivity $K_{SS} = 0.0433$. Also following the development in previous chapters $\frac{1}{R_a} = 1$; $\eta_{tot} K_t = 1.8$; $k_b = 2$; $D = k_f = 0.1$; $\frac{1}{J_{tot}} = \frac{1}{7.226}$; $\frac{r}{i_{tot}} = 0.0615$; and $\rho C_w A v_0 \frac{r}{i_{tot}} = 0.6154$.

a. Substitute these values in the block diagram, and find the transfer function, $T(s) = V(s)/R_v(s)$, using block-diagram reduction rules. [Hint: Start

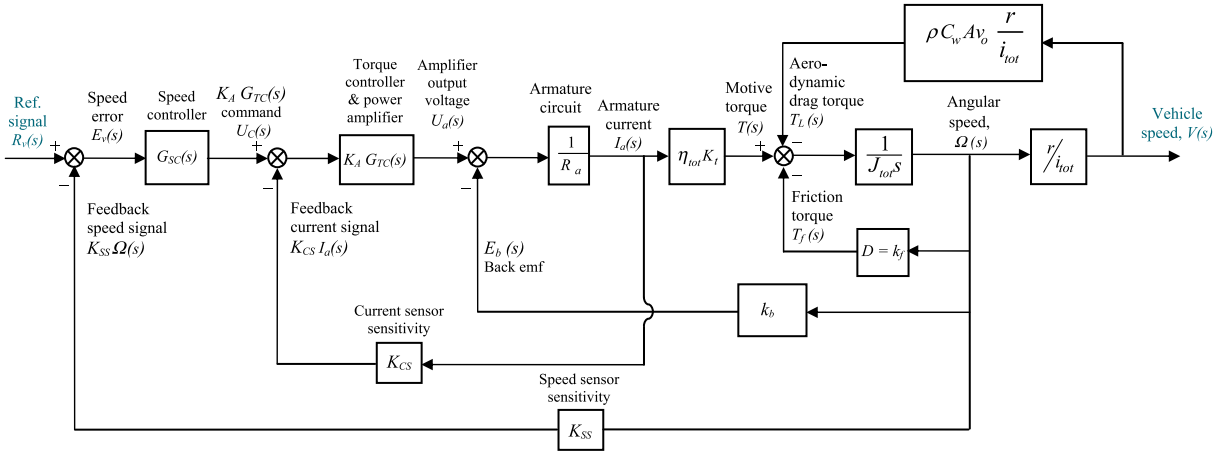


FIGURE P5.54

by moving the last $\frac{r}{i_{tot}}$ block to the right past the pickoff point.]

b. Develop a Simulink model for the original system in Figure P5.54. Set the reference signal input, $r_v(t) = 4\,u(t)$, as a step input with a zero initial value, a step time = 0 seconds, and a final value of 4 volts. Use X-Y graphs to display (over the period from 0 to 8 seconds) the response of the following variables to the step input: (1) change in car speed (m/s), (2) car acceleration (m/s^2), and (3) motor armature current (A).

Simulink SL

MATLAB ML

To record the time and the above three variables (in array format), connect them to four **Workspace** sinks, each of which carry the respective variable name. After the simulation ends, utilize MATLAB plot commands to obtain and edit the three graphs of interest.

Cyber Exploration Laboratory

Experiment 5.1

Objectives To verify the equivalency of the basic forms, including cascade, parallel, and feedback forms. To verify the equivalency of the basic moves, including moving blocks past summing junctions, and moving blocks past pickoff points.

Minimum Required Software Packages MATLAB, Simulink, and the Control System Toolbox

Prelab

1. Find the equivalent transfer function of three cascaded blocks, $G_1(s) = \dfrac{1}{s+1}$. $G_2(s) = \dfrac{1}{s+4}$, and $G_3(s) = \dfrac{s+3}{s+5}$.
2. Find the equivalent transfer function of three parallel blocks, $G_1(s) = \dfrac{1}{s+4}$. $G_2(s) = \dfrac{1}{s+4}$, and $G_3(s) = \dfrac{s+3}{s+5}$.
3. Find the equivalent transfer function of the negative feedback system of Figure P5.55 if $G(s) = \dfrac{s+1}{s(s+2)}$, and $H(s) = \dfrac{s+3}{s+4}$.

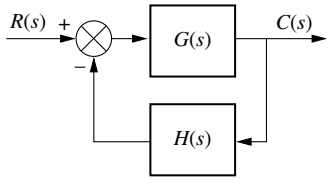


FIGURE P5.55

4. For the system of Prelab 3, push $H(s)$ to the left past the summing junction and draw the equivalent system.
5. For the system of Prelab 3, push $H(s)$ to the right past the pickoff point and draw the equivalent system.

Lab

1. Using Simulink, set up the cascade system of Prelab 1 and the equivalent single block. Make separate plots of the step response of the cascaded system and its equivalent single block. Record the values of settling time and rise time for each step response.
2. Using Simulink, set up the parallel system of Prelab 2 and the equivalent single block. Make separate plots of the step response of the parallel system and its equivalent single block. Record the values of settling time and rise time for each step response.
3. Using Simulink, set up the negative feedback system of Prelab 3 and the equivalent single block. Make separate plots of the step response of the negative feedback system and its equivalent single block. Record the values of settling time and rise time for each step response.
4. Using Simulink, set up the negative feedback systems of Prelabs 3, 4, and 5. Make separate plots of the step response of each of the systems. Record the values of settling time and rise time for each step response.

Postlab

1. Using your lab data, verify the equivalent transfer function of blocks in cascade.
2. Using your lab data, verify the equivalent transfer function of blocks in parallel.
3. Using your lab data, verify the equivalent transfer function of negative feedback systems.
4. Using your lab data, verify the moving of blocks past summing junctions and pickoff points.
5. Discuss your results. Were the equivalencies verified?

Experiment 5.2

Objective To use the various functions within LabVIEW's Control Design and Simulation Module to implement block diagram reduction.

Minimum Required Software Package LabVIEW with the Control Design Simulation Module

Prelab Given the block diagram from Example 5.2, replace $G_1, G_2, G_3, H_1, H_2, H_3$ with the following transfer functions and obtain an equivalent transfer function.

$$G_1 = \frac{1}{s+10}; G_2 = \frac{1}{s+1}; G_3 = \frac{s+1}{s^2+4s+4}; H_1 = \frac{s+1}{s+2}; H_2 = 2; H_3 = 1$$

Lab Use LabVIEW to implement the block diagram from Example 5.2 using the transfer functions given in the Prelab.

Postlab Verify your calculations from the Prelab with that of the equivalent transfer function obtained with LabVIEW.

Experiment 5.3

Objective To use the various functions within LabVIEW's Control Design and Simulation Module and the Mathematics/Polynomial palette to implement Mason's rule for block diagram reduction.

Minimum Required Software Package LabVIEW with Control Design and Simulation Module, Math Script RT Module, and the Mathematics/Polynomial palette.

Prelab Given the block diagram created in the Prelab of Cyber Exploration Laboratory 5.2, use Mason's rule to obtain an equivalent transfer function.

Lab Use LabVIEW's Control Design and Simulation Module as well as the Mathematics/Polynomial functions to implement block diagram reduction using Mason's rule.

Postlab Verify your calculations from the Prelab with that of the equivalent transfer function obtained with LabVIEW.

Bibliography

Åstrom, K., Klein, R. E., and Lennartsson, A. Bicycle Dynamics and Control. *IEEE Control Systems*, August 2005, pp. 26–47.

Bailey, F. N., Cockburn, J. C., and Dee, A. Robust Control for High-Performance Materials Testing. *IEEE Control Systems*, April 1992, pp. 63–70.

Ballard, R. D. *The Discovery of the Titanic*, Warner Books, New York, 1987.

Ben-Dov, D., and Salcudean, S. E. A Force-Controlled Pneumatic Actuator. *IEEE Transactions on Robotics and Automation*, vol. 11, 1995, pp. 906–911.

Cavallo, A., De Maria, G., and Verde, L. Robust Flight Control Systems: A Parameter Space Design. *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, September-October 1992, pp. 1207–1215.

Craig, I. K., Xia, X., and Venter, J. W., Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria, *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

de Vlugt, E., Schouten, A. C., and van derHelm, F.C.T. Adaptation of Reflexive Feedback during Arm Posture to Different Environments. *Biological Cybernetics*, vol. 87, 2002, pp. 10–26.

Evans, J., Krishnamurthy, B., Barrows, B., Skewis, T., and Lumelsky, V. Handling Real-World Motion Planning: A Hospital Transport Robot. *IEEE Control Systems*, February 1992, pp. 15–20.

Graebe, S. F., Goodwin, G. C., and Elsley, G., Control Design and Implementation in Continuous Steel Casting. *IEEE Control Systems*, August 1995, pp. 64–71.

Hostetter, G. H., Savant, C. J., Jr. and Stefani, R. T. *Design of Feedback Control Systems*. 2d ed. Saunders College Publishing, New York, 1989.

Johnson, H. et al. *Unmanned Free-Swimming Submersible (UFSS) System Description*. NRL Memorandum Report 4393. Naval Research Laboratory, Washington, DC, 1980.

Karkoub, M., Her, M.-G., and Chen, J. M. Design and Control of a Haptic Interactive Motion Simulator for Virtual Entertainment Systems, *Robonica*, vol. 28, 2010, pp. 47–56.

Kong, F., and de Keyser, R. Identification and Control of the Mould Level in a Continuous Casting Machine. *Second IEEE Conference on Control Application*, Vancouver, B.C., 1993. pp. 53–58.

Kurland, M., and Papson, T. P. Analog Computer Simulation of Linear Modulation Systems. *Transactions of the Analog/Hybrid Computer Educational Society*, January 1971, pp. 9–18.

Lee, S.-R., Ko, S.-H., Dehbonei, H., Jeon, C.-H., and Kwon, O.-S., Operational Characteristics of PV/Diesel Hybrid Distributed Generation System Using Dual Voltage Source Inverter for Weak Grid, *ISIS 2007 Proceedings,* Eighth Symposium on Advanced Intelligent Systems. 2007.

Lepschy, A. M., Mian, G. A., and Viaro, U. Feedback Control in Ancient Water and Mechanical Clocks. *IEEE Transactions on Education*, vol. 35, 1992, pp. 3–10.

Lin, J.-S., and Kanellakopoulos, I., Nonlinear Design of Active Suspensions. *IEEE Control Systems*, vol. 17, issue 3, June 1997, pp. 45–59.

Mason, S. J. Feedback Theory—Some Properties of Signal-Flow Graphs. *Proc. IRE*, September 1953, pp. 1144–1156.

Milhorn, H. T., Jr. *The Application of Control Theory to Physiological Systems.* W. B. Saunders, Philadelphia, 1966.

Neamen, D. A. *Electronic Circuit Analysis and Design*. McGraw-Hill, 2d ed., 2001, p. 334.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. 1–4.

Piccin, O., Barbé L., Bayle B., and de Mathelin, M. A Force Feedback Teleoperated Needle Insertion Device for Percutaneous Procedures. *Int. J. of Robotics Research*, vol. 28, 2009, p. 1154.

Preitl, Z., Bauer, P., and Bokor, J. A Simple Control Solution for Traction Motor Used in Hybrid Vehicles. *Fourth International Symposium on Applied Computational Intelligence and Informatics.* IEEE, 2007.

Tasch, U., Koontz, J. W., Ignatoski, M. A., and Geselowitz, D. B. An Adaptive Aortic Pressure Observer for the Penn State Electric Ventricular Assist Device. *IEEE Transactions on Biomedical Engineering*, vol. 37, 1990, pp. 374–383.

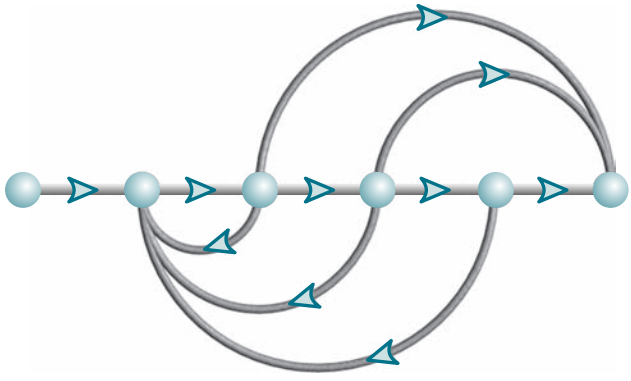
Timothy, L. K., and Bona, B. E. *State Space Analysis: An Introduction*. McGraw-Hill, New York, 1968.

Vaughan, N. D., and Gamble, J. B. The Modeling and Simulation of a Proportional Solenoid Valve. *Journal of Dynamic Systems, Measurements, and Control*, vol. 118, March 1996, pp. 120–125.

Yaniv, Y., Sivan, R., and Landesberg, A. Stability, Controllability and Observability of the ''Four State'' Model for the Sarcomeric Control of Contraction. *Annals of Biomedical Engineering*, vol. 34, 2006, pp. 778–789.

Stability

6



Chapter Learning Outcomes

After completing this chapter the student will be able to:

- Make and interpret a basic Routh table to determine the stability of a system (Sections 6.1–6.2)
- Make and interpret a Routh table where either the first element of a row is zero or an entire row is zero (Sections 6.3–6.4)
- Use a Routh table to determine the stability of a system represented in state space (Section 6.5)

State Space
SS

Case Study Learning Outcomes

You will be able to demonstrate your knowledge of the chapter objectives with case studies as follows:

- Given the antenna azimuth position control system shown on the front endpapers, you will be able to find the range of preamplifier gain to keep the system stable.
- Given the block diagrams for the UFSS vehicle's pitch and heading control systems on the back endpapers, you will be able to determine the range of gain for stability of the pitch or heading control system.

6.1 Introduction

In Chapter 1, we saw that three requirements enter into the design of a control system: transient response, stability, and steady-state errors. Thus far we have covered transient response, which we will revisit in Chapter 8. We are now ready to discuss the next requirement, stability.

Stability is the most important system specification. If a system is unstable, transient response and steady-state errors are moot points. An unstable system cannot be designed for a specific transient response or steady-state error requirement. What, then, is stability? There are many definitions for stability, depending upon the kind of system or the point of view. In this section, we limit ourselves to linear, time-invariant systems.

In Section 1.5, we discussed that we can control the output of a system if the steady-state response consists of only the forced response. But the total response of a system is the sum of the forced and natural responses, or

$$c(t) = c_{\text{forced}}(t) + c_{\text{natural}}(t) \tag{6.1}$$

Using these concepts, we present the following definitions of stability, instability, and marginal stability:

A linear, time-invariant system is *stable* if the natural response approaches zero as time approaches infinity.

A linear, time-invariant system is *unstable* if the natural response grows without bound as time approaches infinity.

A linear, time-invariant system is *marginally stable* if the natural response neither decays nor grows but remains constant or oscillates as time approaches infinity.

Thus, the definition of stability implies that only the forced response remains as the natural response approaches zero.

These definitions rely on a description of the natural response. When one is looking at the total response, it may be difficult to separate the natural response from the forced response. However, we realize that if the input is bounded and the total response is not approaching infinity as time approaches infinity, then the natural response is obviously not approaching infinity. If the input is unbounded, we see an unbounded total response, and we cannot arrive at any conclusion about the stability of the system; we cannot tell whether the total response is unbounded because the forced response is unbounded or because the natural response is unbounded. Thus, our alternate definition of *stability*, one that regards the total response and implies the first definition based upon the natural response, is this:

A system is stable if *every* bounded input yields a bounded output.

We call this statement the bounded-input, bounded-output (BIBO) definition of stability.

Let us now produce an alternate definition for instability based on the total response rather than the natural response. We realize that if the input is bounded but the total response is unbounded, the system is unstable, since we can conclude that the natural response approaches infinity as time approaches infinity. If the input is unbounded, we will see an unbounded total response, and we cannot draw any conclusion about the stability of the system; we cannot tell whether the total response is unbounded because the forced response is unbounded or because the

natural response is unbounded. Thus, our alternate definition of *instability*, one that regards the total response, is this:

A system is unstable if *any* bounded input yields an unbounded output.

These definitions help clarify our previous definition of *marginal stability*, which really means that the system is stable for some bounded inputs and unstable for others. For example, we will show that if the natural response is undamped, a bounded sinusoidal input of the same frequency yields a natural response of growing oscillations. Hence, the system appears stable for all bounded inputs except this one sinusoid. Thus, marginally stable systems by the natural response definitions are included as unstable systems under the BIBO definitions.

Let us summarize our definitions of stability for linear, time-invariant systems. Using the natural response:

1. A system is stable if the natural response approaches zero as time approaches infinity.
2. A system is unstable if the natural response approaches infinity as time approaches infinity.
3. A system is marginally stable if the natural response neither decays nor grows but remains constant or oscillates.

Using the total response (BIBO):

1. A system is stable if *every* bounded input yields a bounded output.
2. A system is unstable if *any* bounded input yields an unbounded output.

Physically, an unstable system whose natural response grows without bound can cause damage to the system, to adjacent property, or to human life. Many times systems are designed with limited stops to prevent total runaway. From the perspective of the time response plot of a physical system, instability is displayed by transients that grow without bound and, consequently, a total response that does not approach a steady-state value or other forced response.[1]

How do we determine if a system is stable? Let us focus on the natural response definitions of stability. Recall from our study of system poles that poles in the left half-plane (lhp) yield either pure exponential decay or damped sinusoidal natural responses. These natural responses decay to zero as time approaches infinity. Thus, if the closed-loop system poles are in the left half of the plane and hence have a negative real part, the system is stable. That is, *stable systems have closed-loop transfer functions with poles only in the left half-plane.*

Poles in the right half-plane (rhp) yield either pure exponentially increasing or exponentially increasing sinusoidal natural responses. These natural responses approach infinity as time approaches infinity. Thus, if the closed-loop system poles are in the right half of the s-plane and hence have a positive real part, the system is unstable. Also, poles of multiplicity greater than 1 on the imaginary axis lead to the sum of responses of the form $At^n \cos(\omega t + \phi)$, where $n = 1, 2, \ldots$, which also approaches infinity as time approaches infinity. Thus, *unstable systems have closed-loop transfer functions with at least one pole in the right half-plane and/or poles of multiplicity greater than 1 on the imaginary axis.*

[1] Care must be taken here to distinguish between natural responses growing without bound and a forced response, such as a ramp or exponential increase, that also grows without bound. A system whose forced response approaches infinity is stable as long as the natural response approaches zero.

Finally, a system that has imaginary axis poles of multiplicity 1 yields pure sinusoidal oscillations as a natural response. These responses neither increase nor decrease in amplitude. Thus, *marginally stable systems have closed-loop transfer functions with only imaginary axis poles of multiplicity 1 and poles in the left half-plane.*

As an example, the unit step response of the stable system of Figure 6.1(*a*) is compared to that of the unstable system of Figure 6.1(*b*). The responses, also shown in Figure 6.1, show that while the oscillations for the stable system diminish, those for the unstable system increase without bound. Also notice that the stable system's response in this case approaches a steady-state value of unity.

It is not always a simple matter to determine if a feedback control system is stable. Unfortunately, a typical problem that arises is shown in Figure 6.2. Although we know the poles of the forward transfer function in Figure 6.2(*a*), we do not know the location of the poles of the equivalent closed-loop system of Figure 6.2(*b*) without factoring or otherwise solving for the roots.

However, under certain conditions, we can draw some conclusions about the stability of the system. First, if the closed-loop transfer function has only

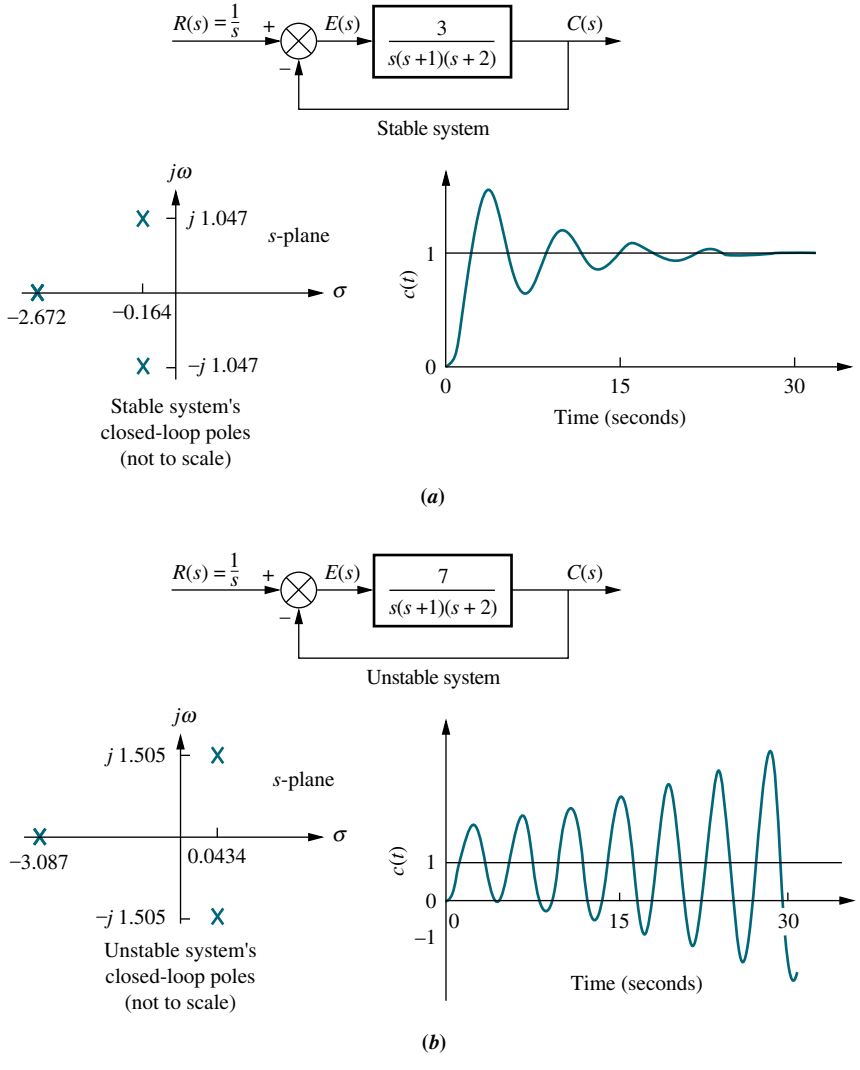


FIGURE 6.1 Closed-loop poles and response: **a.** stable system; **b.** unstable system

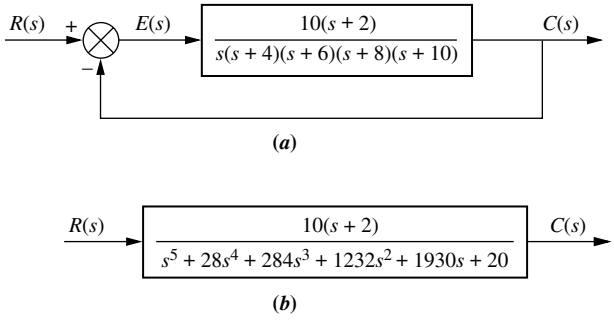


FIGURE 6.2 Common cause of problems in finding closed-loop poles: **a.** original system; **b.** equivalent system

left–half-plane poles, then the factors of the denominator of the closed-loop system transfer function consist of products of terms such as $(s + a_i)$, where a_i is real and positive, or complex with a positive real part. The product of such terms is a polynomial with all positive coefficients.[2] No term of the polynomial can be missing, since that would imply cancellation between positive and negative coefficients or imaginary axis roots in the factors, which is not the case. Thus, a sufficient condition for a system to be unstable is that all signs of the coefficients of the denominator of the closed-loop transfer function are not the same. If powers of s are missing, the system is either unstable or, at best, marginally stable. Unfortunately, if all coefficients of the denominator are positive and not missing, we do not have definitive information about the system's pole locations.

If the method described in the previous paragraph is not sufficient, then a computer can be used to determine the stability by calculating the root locations of the denominator of the closed-loop transfer function. Today some hand-held calculators can evaluate the roots of a polynomial. There is, however, another method to test for stability without having to solve for the roots of the denominator. We discuss this method in the next section.

6.2 Routh-Hurwitz Criterion

In this section, we learn a method that yields stability information without the need to solve for the closed-loop system poles. Using this method, we can tell how many closed-loop system poles are in the left half-plane, in the right half-plane, and on the $j\omega$-axis. (Notice that we say *how many*, not *where*.) We can find the number of poles in each section of the s-plane, but we cannot find their coordinates. The method is called the *Routh-Hurwitz criterion* for stability (*Routh*, *1905*).

The method requires two steps: (1) Generate a data table called a *Routh table* and (2) interpret the Routh table to tell how many closed-loop system poles are in the left half-plane, the right half-plane, and on the $j\omega$-axis. You might wonder why we study the Routh-Hurwitz criterion when modern calculators and computers can tell us the exact location of system poles. The power of the method lies in design rather than analysis. For example, if you have an unknown parameter in the denominator of a transfer function, it is difficult to determine via a calculator the range of this parameter to yield stability. You would probably rely on trial and error to answer the

[2] The coefficients can also be made all negative by multiplying the polynomial by −1. This operation does not change the root location.

stability question. We shall see later that the Routh-Hurwitz criterion can yield a closed-form expression for the range of the unknown parameter.

In this section, we make and interpret a basic Routh table. In the next section, we consider two special cases that can arise when generating this data table.

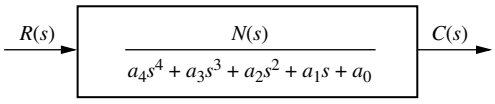


FIGURE 6.3 Equivalent closed-loop transfer function

Generating a Basic Routh Table

Look at the equivalent closed-loop transfer function shown in Figure 6.3. Since we are interested in the system poles, we focus our attention on the denominator. We first create the Routh table shown in Table 6.1. Begin by labeling the rows with powers of s from the highest power of the denominator of the closed-loop transfer function to s^0. Next start with the coefficient of the highest power of s in the denominator and list, horizontally in the first row, every other coefficient. In the second row, list horizontally, starting with the next highest power of s, every coefficient that was skipped in the first row.

The remaining entries are filled in as follows. Each entry is a negative determinant of entries in the previous two rows divided by the entry in the first column directly above the calculated row. The left-hand column of the determinant is always the first column of the previous two rows, and the right-hand column is the elements of the column above and to the right. The table is complete when all of the rows are completed down to s^0. Table 6.2 is the completed Routh table. Let us look at an example.

TABLE 6.1 Initial layout for Routh table

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2			
s^1			
s^0			

TABLE 6.2 Completed Routh table

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$-\dfrac{\begin{vmatrix} a_4 & a_2 \\ a_3 & a_1 \end{vmatrix}}{a_3} = b_1$	$-\dfrac{\begin{vmatrix} a_4 & a_0 \\ a_3 & 0 \end{vmatrix}}{a_3} = b_2$	$-\dfrac{\begin{vmatrix} a_4 & 0 \\ a_3 & 0 \end{vmatrix}}{a_3} = 0$
s^1	$-\dfrac{\begin{vmatrix} a_3 & a_1 \\ b_1 & b_2 \end{vmatrix}}{b_1} = c_1$	$-\dfrac{\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$	$-\dfrac{\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$
s^0	$-\dfrac{\begin{vmatrix} b_1 & b_2 \\ c_1 & 0 \end{vmatrix}}{c_1} = d_1$	$-\dfrac{\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$	$-\dfrac{\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$

Example 6.1

Creating a Routh Table

PROBLEM: Make the Routh table for the system shown in Figure 6.4(a).

SOLUTION: The first step is to find the equivalent closed-loop system because we want to test the denominator of this function, not the given forward transfer

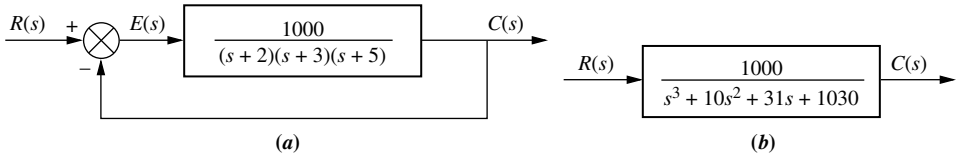


FIGURE 6.4 **a.** Feedback system for Example 6.1; **b.** equivalent closed-loop system

TABLE 6.3 Completed Routh table for Example 6.1

s^3	1	31	0
s^2	$\cancel{10}\ \ 1$	$\cancel{1030}\ \ 103$	0
s^1	$\dfrac{-\begin{vmatrix} 1 & 31 \\ 1 & 103 \end{vmatrix}}{1} = -72$	$\dfrac{-\begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix}}{1} = 0$	$\dfrac{-\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}}{1} = 0$
s^0	$\dfrac{-\begin{vmatrix} 1 & 103 \\ -72 & 0 \end{vmatrix}}{-72} = 103$	$\dfrac{-\begin{vmatrix} 1 & 0 \\ -72 & 0 \end{vmatrix}}{-72} = 0$	$\dfrac{-\begin{vmatrix} 1 & 0 \\ -72 & 0 \end{vmatrix}}{-72} = 0$

function, for pole location. Using the feedback formula, we obtain the equivalent system of Figure 6.4(b). The Routh-Hurwitz criterion will be applied to this denominator. First label the rows with powers of s from s^3 down to s^0 in a vertical column, as shown in Table 6.3. Next form the first row of the table, using the coefficients of the denominator of the closed-loop transfer function. Start with the coefficient of the highest power and skip every other power of s. Now form the second row with the coefficients of the denominator skipped in the previous step. Subsequent rows are formed with determinants, as shown in Table 6.2.

For convenience, any row of the Routh table can be multiplied by a positive constant without changing the values of the rows below. This can be proved by examining the expressions for the entries and verifying that any multiplicative constant from a previous row cancels out. In the second row of Table 6.3, for example, the row was multiplied by 1/10. We see later that care must be taken not to multiply the row by a negative constant.

Interpreting the Basic Routh Table

Now that we know how to generate the Routh table, let us see how to interpret it. The basic Routh table applies to systems with poles in the left and right half-planes. Systems with imaginary poles and the kind of Routh table that results will be discussed in the next section. Simply stated, the Routh-Hurwitz criterion declares that *the number of roots of the polynomial that are in the right half-plane is equal to the number of sign changes in the first column*.

If the closed-loop transfer function has all poles in the left half of the s-plane, the system is stable. Thus, a system is stable if there are no sign changes in the first column of the Routh table. For example, Table 6.3 has two sign changes in the first column. The first sign change occurs from 1 in the s^2 row to -72 in the s^1 row. The second occurs from -72 in the s^1 row to 103 in the s^0 row. Thus, the system of Figure 6.4 is unstable since two poles exist in the right half-plane.

Skill-Assessment Exercise 6.1

PROBLEM: Make a Routh table and tell how many roots of the following polynomial are in the right half-plane and in the left half-plane.

$$P(s) = 3s^7 + 9s^6 + 6s^5 + 4s^4 + 7s^3 + 8s^2 + 2s + 6$$

ANSWER: Four in the right half-plane (rhp), three in the left half-plane (lhp).

The complete solution is at www.wiley.com/college/nise.

WileyPLUS
WPCS
Control Solutions

Now that we have described how to generate and interpret a basic Routh table, let us look at two special cases that can arise.

6.3 Routh-Hurwitz Criterion: Special Cases

Two special cases can occur: (1) The Routh table sometimes will have a zero *only in the first column* of a row, or (2) the Routh table sometimes will have an *entire row* that consists of zeros. Let us examine the first case.

Zero Only in the First Column

If the first element of a row is zero, division by zero would be required to form the next row. To avoid this phenomenon, an epsilon, ϵ, is assigned to replace the zero in the first column. The value ϵ is then allowed to approach zero from either the positive or the negative side, after which the signs of the entries in the first column can be determined. Let us look at an example.

Example 6.2

Stability via Epsilon Method

TryIt 6.1

Use the following MATLAB statement to find the poles of the closed-loop transfer function in Eq. (6.2).

```
roots([1 2 3 6 5 3])
```

PROBLEM: Determine the stability of the closed-loop transfer function

$$T(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3} \tag{6.2}$$

SOLUTION: The solution is shown in Table 6.4. We form the Routh table by using the denominator of Eq. (6.2). Begin by assembling the Routh table down to the row where a zero appears *only* in the first column (the s^3 row). Next replace the zero by a small number, ϵ, and complete the table. To begin the interpretation, we must first assume a sign, positive or negative, for the quantity ϵ. Table 6.5 shows the first column of Table 6.4 along with the resulting signs for choices of ϵ positive and ϵ negative.

TABLE 6.4 Completed Routh table for Example 6.2

s^5	1	3	5
s^4	2	6	3
s^3	~~0~~ ϵ	$\frac{7}{2}$	0
s^2	$\frac{6\epsilon - 7}{\epsilon}$	3	0
s^1	$\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	0	0
s^0	3	0	0

TABLE 6.5 Determining signs in first column of a Routh table with zero as first element in a row

Label	First column	$\epsilon = +$	$\epsilon = -$
s^5	1	+	+
s^4	2	+	+
s^3	~~0~~ ϵ	+	−
s^2	$\frac{6\epsilon - 7}{\epsilon}$	−	+
s^1	$\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	+	+
s^0	3	+	+

If ϵ is chosen positive, Table 6.5 will show a sign change from the s^3 row to the s^2 row, and there will be another sign change from the s^2 row to the s^1 row. Hence, the system is unstable and has two poles in the right half-plane.

Alternatively, we could choose ϵ negative. Table 6.5 would then show a sign change from the s^4 row to the s^3 row. Another sign change would occur from the s^3 row to the s^2 row. Our result would be exactly the same as that for a positive choice for ϵ. Thus, the system is unstable, with two poles in the right half-plane.

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch6sp1 in Appendix F at www.wiley.com/college/nise. You will learn how to use the Symbolic Math Toolbox to calculate the values of cells in a Routh table even if the table contains symbolic objects, such as ϵ. You will see that the Symbolic Math Toolbox and MATLAB yield an alternate way to generate the Routh table for Example 6.2.

Another method that can be used when a zero appears only in the first column of a row is derived from the fact that a polynomial that has the reciprocal roots of the original polynomial has its roots distributed the same—right half-plane, left half-plane, or imaginary axis—because taking the reciprocal of the root value does not move it to another region. Thus, if we can find the polynomial that has the reciprocal roots of the original, it is possible that the Routh table for the new polynomial will not have a zero in the first column. This method is usually computationally easier than the epsilon method just described.

We now show that the polynomial we are looking for, the one with the reciprocal roots, is simply the original polynomial with its coefficients written in reverse order (*Phillips, 1991*). Assume the equation

$$s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = 0 \tag{6.3}$$

If s is replaced by $1/d$, then d will have roots which are the reciprocal of s. Making this substitution in Eq. (6.3),

$$\left(\frac{1}{d}\right)^n + a_{n-1}\left(\frac{1}{d}\right)^{n-1} + \cdots + a_1\left(\frac{1}{d}\right) + a_0 = 0 \tag{6.4}$$

Factoring out $(1/d)^n$,

$$\left(\frac{1}{d}\right)^n \left[1 + a_{n-1}\left(\frac{1}{d}\right)^{-1} + \cdots + a_1\left(\frac{1}{d}\right)^{(1-n)} + a_0\left(\frac{1}{d}\right)^{-n}\right]$$

$$= \left(\frac{1}{d}\right)^n [1 + a_{n-1}d + \cdots + a_1 d^{(n-1)} + a_0 d^n] = 0 \tag{6.5}$$

Thus, the polynomial with reciprocal roots is a polynomial with the coefficients written in reverse order. Let us redo the previous example to show the computational advantage of this method.

Example 6.3

Stability via Reverse Coefficients

PROBLEM: Determine the stability of the closed-loop transfer function

$$T(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3} \tag{6.6}$$

SOLUTION: First write a polynomial that has the reciprocal roots of the denominator of Eq. (6.6). From our discussion, this polynomial is formed by writing the denominator of Eq. (6.6) in reverse order. Hence,

$$D(s) = 3s^5 + 5s^4 + 6s^3 + 3s^2 + 2s + 1 \tag{6.7}$$

We form the Routh table as shown in Table 6.6 using Eq. (6.7). Since there are two sign changes, the system is unstable and has two right-half-plane poles. This is the same as the result obtained in Example 6.2. Notice that Table 6.6 does not have a zero in the first column.

TABLE 6.6 Routh table for Example 6.3

s^5	3	6	2
s^4	5	3	1
s^3	4.2	1.4	
s^2	1.33	1	
s^1	−1.75		
s^0	1		

Entire Row is Zero

We now look at the second special case. Sometimes while making a Routh table, we find that an entire row consists of zeros because there is an even polynomial that is a factor of the original polynomial. This case must be handled differently from the case of a zero in only the first column of a row. Let us look at an example that demonstrates how to construct and interpret the Routh table when an entire row of zeros is present.

Example 6.4

Stability via Routh Table with Row of Zeros

PROBLEM: Determine the number of right-half-plane poles in the closed-loop transfer function

$$T(s) = \frac{10}{s^5 + 7s^4 + 6s^3 + 42s^2 + 8s + 56} \tag{6.8}$$

SOLUTION: Start by forming the Routh table for the denominator of Eq. (6.8) (see Table 6.7). At the second row we multiply through by 1/7 for convenience. We stop at the third row, since the entire row consists of zeros, and use the following

TABLE 6.7 Routh table for Example 6.4

s^5			1			6			8
s^4		~~7~~	1		~~42~~	6		~~56~~	8
s^3	~~0~~	~~4~~	1	~~0~~	~~12~~	3	~~0~~	~~0~~	0
s^2			3			8			0
s^1			$\frac{1}{3}$			0			0
s^0			8			0			0

procedure. First we return to the row immediately above the row of zeros and form an auxiliary polynomial, using the entries in that row as coefficients. The polynomial will start with the power of s in the label column and continue by skipping every other power of s. Thus, the polynomial formed for this example is

$$P(s) = s^4 + 6s^2 + 8 \tag{6.9}$$

Next we differentiate the polynomial with respect to s and obtain

$$\frac{dP(s)}{ds} = 4s^3 + 12s + 0 \tag{6.10}$$

Finally, we use the coefficients of Eq. (6.10) to replace the row of zeros. Again, for convenience, the third row is multiplied by 1/4 after replacing the zeros.

The remainder of the table is formed in a straightforward manner by following the standard form shown in Table 6.2. Table 6.7 shows that all entries in the first column are positive. Hence, there are no right–half-plane poles.

Let us look further into the case that yields an entire row of zeros. An entire row of zeros will appear in the Routh table when a purely even or purely odd polynomial is a factor of the original polynomial. For example, $s^4 + 5s^2 + 7$ is an even polynomial; it has only even powers of s. Even polynomials only have roots that are symmetrical about the origin.[3] This symmetry can occur under three conditions of root position: (1) The roots are symmetrical and real, (2) the roots are symmetrical and imaginary, or (3) the roots are quadrantal. Figure 6.5 shows examples of these cases. Each case or combination of these cases will generate an even polynomial.

It is this even polynomial that causes the row of zeros to appear. Thus, the row of zeros tells us of the existence of an even polynomial whose roots are symmetric about the origin. Some of these roots could be on the $j\omega$-axis. On the other hand, since $j\omega$ roots are symmetric about the origin, if we do not have a row of zeros, we cannot possibly have $j\omega$ roots.

Another characteristic of the Routh table for the case in question is that the row previous to the row of zeros contains the even polynomial that is a factor of the original polynomial. Finally, everything from the row containing the even polynomial down to the end of the Routh table is a test of only the even polynomial. Let us put these facts together in an example.

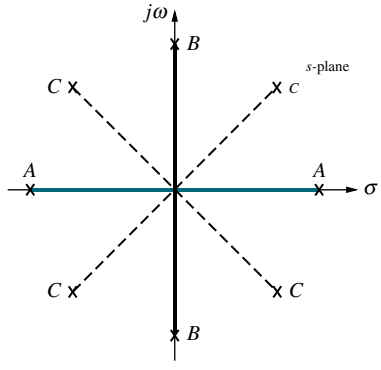


FIGURE 6.5 Root positions to generate even polynomials: A, B, C, or any combination

[3] The polynomial $s^5 + 5s^3 + 7s$ is an example of an odd polynomial; it has only odd powers of s. Odd polynomials are the product of an even polynomial and an odd power of s. Thus, the constant term of an odd polynomial is always missing.

Example 6.5

Pole Distribution via Routh Table with Row of Zeros

PROBLEM: For the transfer function

$$T(s) = \frac{20}{s^8 + s^7 + 12s^6 + 22s^5 + 39s^4 + 59s^3 + 48s^2 + 38s + 20} \tag{6.11}$$

tell how many poles are in the right half-plane, in the left half-plane, and on the $j\omega$-axis.

SOLUTION: Use the denominator of Eq. (6.11) and form the Routh table in Table 6.8. For convenience the s^6 row is multiplied by 1/10, and the s^5 row is multiplied by 1/20. At the s^3 row we obtain a row of zeros. Moving back one row to s^4, we extract the even polynomial, $P(s)$, as

$$P(s) = s^4 + 3s^2 + 2 \tag{6.12}$$

TABLE 6.8 Routh table for Example 6.5

s^8	1	12	39	48	20
s^7	1	22	59	38	0
s^6	$-\cancel{10} - 1$	$-\cancel{20} - 2$	$\cancel{10}$ 1	$\cancel{20}$ 2	0
s^5	$\cancel{20}$ 1	$\cancel{60}$ 3	$\cancel{40}$ 2	0	0
s^4	1	3	2	0	0
s^3	$\cancel{0}$ $\cancel{4}$ 2	$\cancel{0}$ $\cancel{6}$ 3	$\cancel{0}$ $\cancel{0}$ 0	0	0
s^2	$\frac{3}{\cancel{2}}$ 3	$\cancel{2}$ 4	0	0	0
s^1	$\frac{1}{3}$	0	0	0	0
s^0	4	0	0	0	0

This polynomial will divide evenly into the denominator of Eq. (6.11) and thus is a factor. Taking the derivative with respect to s to obtain the coefficients that replace the row of zeros in the s^3 row, we find

$$\frac{dP(s)}{ds} = 4s^3 + 6s + 0 \tag{6.13}$$

Replace the row of zeros with 4, 6, and 0 and multiply the row by 1/2 for convenience. Finally, continue the table to the s^0 row, using the standard procedure.

How do we now interpret this Routh table? Since all entries from the even polynomial at the s^4 row down to the s^0 row are a test of the even polynomial, we begin to draw some conclusions about the roots of the even polynomial. No sign changes exist from the s^4 row down to the s^0 row. Thus, the even polynomial does not have right–half-plane poles. Since there are no right–half-plane poles, no left–half-plane poles are present because of the requirement for symmetry. Hence, the even polynomial, Eq. (6.12), must have all four of its poles on the $j\omega$-axis.[4] These results are summarized in the first column of Table 6.9.

[4] A necessary condition for stability is that the $j\omega$ roots have unit multiplicity. The even polynomial must be checked for multiple $j\omega$ roots. For this case, the existence of multiple $j\omega$ roots would lead to a perfect, fourth-order square polynomial. Since Eq. (6.12) is not a perfect square, the four $j\omega$ roots are distinct.

TABLE 6.9 Summary of pole locations for Example 6.5

	Polynomial		
Location	**Even (fourth-order)**	**Other (fourth-order)**	**Total (eighth-order)**
Right half-plane	0	2	2
Left half-plane	0	2	2
$j\omega$	4	0	4

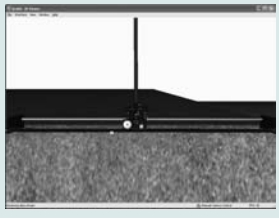
The remaining roots of the total polynomial are evaluated from the s^8 row down to the s^4 row. We notice two sign changes: one from the s^7 row to the s^6 row and the other from the s^6 row to the s^5 row. Thus, the other polynomial must have two roots in the right half-plane. These results are included in Table 6.9 under "Other". The final tally is the sum of roots from each component, the even polynomial and the other polynomial, as shown under "Total" in Table 6.9. Thus, the system has two poles in the right half-plane, two poles in the left half-plane, and four poles on the $j\omega$-axis; it is unstable because of the right–half-plane poles.

We now summarize what we have learned about polynomials that generate entire rows of zeros in the Routh table. These polynomials have a purely even factor with roots that are symmetrical about the origin. The even polynomial appears in the Routh table in the row directly above the row of zeros. Every entry in the table from the even polynomial's row to the end of the chart applies only to the even polynomial. Therefore, the number of sign changes from the even polynomial to the end of the table equals the number of right-half-plane roots of the even polynomial. Because of the symmetry of roots about the origin, the even polynomial must have the same number of left–half-plane roots as it does right–half-plane roots. Having accounted for the roots in the right and left half-planes, we know the remaining roots must be on the $j\omega$-axis.

Every row in the Routh table from the beginning of the chart to the row containing the even polynomial applies only to the other factor of the original polynomial. For this factor, the number of sign changes, from the beginning of the table down to the even polynomial, equals the number of right–half-plane roots. The remaining roots are left–half-plane roots. There can be no $j\omega$ roots contained in the other polynomial.

Virtual Experiment 6.1
Stability

Put theory into practice and evaluate the stability of the Quanser Linear Inverted Pendulum in LabVIEW. When in the upward balanced position, this system addresses the challenge of stabilizing a rocket during take-off. In the downward position it emulates the construction gantry crane.



Virtual experiments are found on WileyPLUS.

Skill-Assessment Exercise 6.2

PROBLEM: Use the Routh-Hurwitz criterion to find how many poles of the following closed-loop system, $T(s)$, are in the rhp, in the lhp, and on the $j\omega$-axis:

$$T(s) = \frac{s^3 + 7s^2 - 21s + 10}{s^6 + s^5 - 6s^4 + 0s^3 - s^2 - s + 6}$$

ANSWER: Two rhp, two lhp, and two $j\omega$

The complete solution is at www.wiley.com/college/nise.

Let us demonstrate the usefulness of the Routh-Hurwitz criterion with a few additional examples.

6.4 Routh-Hurwitz Criterion: Additional Examples

The previous two sections have introduced the Routh-Hurwitz criterion. Now we need to demonstrate the method's application to a number of analysis and design problems.

Example 6.6

Standard Routh-Hurwitz

PROBLEM: Find the number of poles in the left half-plane, the right half-plane, and on the $j\omega$-axis for the system of Figure 6.6.

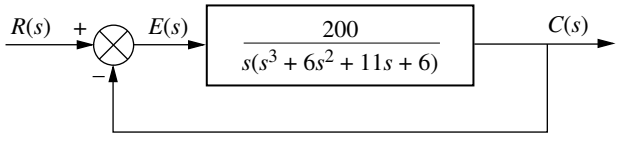


FIGURE 6.6 Feedback control system for Example 6.6

SOLUTION: First, find the closed-loop transfer function as

$$T(s) = \frac{200}{s^4 + 6s^3 + 11s^2 + 6s + 200} \tag{6.14}$$

The Routh table for the denominator of Eq. (6.14) is shown as Table 6.10. For clarity, we leave most zero cells blank. At the s^1 row there is a negative coefficient; thus, there are two sign changes. The system is unstable, since it has two right–half-plane poles and two left–half-plane poles. The system cannot have $j\omega$ poles since a row of zeros did not appear in the Routh table.

TABLE 6.10 Routh table for Example 6.6

s^4		1	11	200
s^3	~~6~~	1	~~6~~ 1	
s^2	~~10~~	1	~~200~~ 20	
s^1		−19		
s^0		20		

The next example demonstrates the occurrence of a zero in only the first column of a row.

Example 6.7

Routh-Hurwitz with Zero in First Column

PROBLEM: Find the number of poles in the left half-plane, the right half-plane, and on the $j\omega$-axis for the system of Figure 6.7.

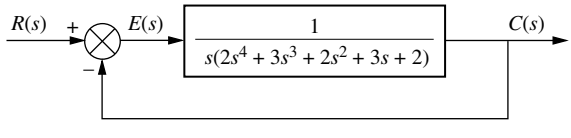


FIGURE 6.7 Feedback control system for Example 6.7

SOLUTION: The closed-loop transfer function is

$$T(s) = \frac{1}{2s^5 + 3s^4 + 2s^3 + 3s^2 + 2s + 1} \tag{6.15}$$

Form the Routh table shown as Table 6.11, using the denominator of Eq. (6.15). A zero appears in the first column of the s^3 row. Since the entire row is not zero, simply replace the zero with a small quantity, ϵ, and continue the table. Permitting ϵ to be a small, positive quantity, we find that the first term of the s^2 row is negative. Thus, there are two sign changes, and the system is unstable, with two poles in the right half-plane. The remaining poles are in the left half-plane.

TABLE 6.11 Routh table for Example 6.7

s^5	2	2	2
s^4	3	3	1
s^3	~~0~~ ϵ	$\frac{4}{3}$	
s^2	$\frac{3\epsilon - 4}{\epsilon}$	1	
s^1	$\frac{12\epsilon - 16 - 3\epsilon^2}{9\epsilon - 12}$		
s^0	1		

We also can use the alternative approach, where we produce a polynomial whose roots are the reciprocal of the original. Using the denominator of Eq. (6.15), we form a polynomial by writing the coefficients in reverse order,

$$s^5 + 2s^4 + 3s^3 + 2s^2 + 3s + 2 \tag{6.16}$$

The Routh table for this polynomial is shown as Table 6.12. Unfortunately, in this case we also produce a zero only in the first column at the s^2 row. However, the table is easier to work with than Table 6.11. Table 6.12 yields the same results as Table 6.11: three poles in the left half-plane and two poles in the right half-plane. The system is unstable.

TABLE 6.12 Alternative Routh table for Example 6.7

s^5	1	3	3
s^4	2	2	2
s^3	2	2	
s^2	~~0~~ ϵ	2	
s^1	$\frac{2\epsilon - 4}{\epsilon}$		
s^0	2		

MATLAB
ML

Students who are using MATLAB should now run ch6p1 in Appendix B. You will learn how to perform block diagram reduction to find $T(s)$, followed by an evaluation of the closed-loop system's poles to determine stability. This exercise uses MATLAB to do Example 6.7.

In the next example, we see an entire row of zeros appear along with the possibility of imaginary roots.

Example 6.8

Routh-Hurwitz with Row of Zeros

PROBLEM: Find the number of poles in the left half-plane, the right half-plane, and on the $j\omega$-axis for the system of Figure 6.8. Draw conclusions about the stability of the closed-loop system.

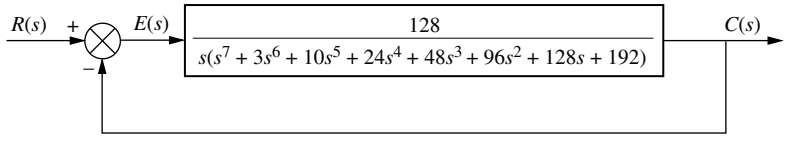


FIGURE 6.8 Feedback control system for Example 6.8

TryIt 6.2

Use MATLAB, The Control System Toolbox, and the following statements to find the closed-loop transfer function, $T(s)$, for Figure 6.8 and the closed-loop poles.

```
numg=128;
deng=[1 3 10 24 ...
 48 96 128 192 0];
G=tf(numg,deng);
T=feedback(G,1)
poles=pole(T)
```

SOLUTION: The closed-loop transfer function for the system of Figure 6.8 is

$$T(s) = \frac{128}{s^8 + 3s^7 + 10s^6 + 24s^5 + 48s^4 + 96s^3 + 128s^2 + 192s + 128} \tag{6.17}$$

Using the denominator, form the Routh table shown as Table 6.13. A row of zeros appears in the s^5 row. Thus, the closed-loop transfer function denominator must have an even polynomial as a factor. Return to the s^6 row and form the even polynomial:

$$P(s) = s^6 + 8s^4 + 32s^2 + 64 \tag{6.18}$$

TABLE 6.13 Routh table for Example 6.8

s^8	1	10	48	128	128
s^7	~~3~~ 1	~~24~~ 8	~~96~~ 32	~~192~~ 64	
s^6	~~2~~ 1	~~16~~ 8	~~64~~ 32	~~128~~ 64	
s^5	~~0~~ ~~6~~ 3	~~0~~ ~~32~~ 16	~~0~~ ~~64~~ 32	~~0~~ ~~0~~ 0	
s^4	$\frac{8}{3}$ 1	$\frac{64}{3}$ 8	~~64~~ 24		
s^3	~~8~~ − 1	~~40~~ − 5			
s^2	~~3~~ 1	~~24~~ 8			
s^1	3				
s^0	8				

Differentiate this polynomial with respect to s to form the coefficients that will replace the row of zeros:

$$\frac{dP(s)}{ds} = 6s^5 + 32s^3 + 64s + 0 \tag{6.19}$$

Replace the row of zeros at the s^5 row by the coefficients of Eq. (6.19) and multiply through by 1/2 for convenience. Then complete the table.

We note that there are two sign changes from the even polynomial at the s^6 row down to the end of the table. Hence, the even polynomial has two right–half-

TABLE 6.14 Summary of pole locations for Example 6.8

	Polynomial		
Location	**Even (sixth-order)**	**Other (second-order)**	**Total (eighth-order)**
Right half-plane	2	0	2
Left half-plane	2	2	4
$j\omega$	2	0	2

plane poles. Because of the symmetry about the origin, the even polynomial must have an equal number of left–half-plane poles. Therefore, the even polynomial has two left–half-plane poles. Since the even polynomial is of sixth order, the two remaining poles must be on the $j\omega$-axis.

There are no sign changes from the beginning of the table down to the even polynomial at the s^6 row. Therefore, the rest of the polynomial has no right–half-plane poles. The results are summarized in Table 6.14. The system has two poles in the right half-plane, four poles in the left half-plane, and two poles on the $j\omega$-axis, which are of unit multiplicity. The closed-loop system is unstable because of the right–half-plane poles.

The Routh-Hurwitz criterion gives vivid proof that changes in the gain of a feedback control system result in differences in transient response because of changes in closed-loop pole locations. The next example demonstrates this concept. We will see that for control systems, such as those shown in Figure 6.9, gain variations can move poles from stable regions of the s-plane onto the $j\omega$-axis and then into the right half-plane.

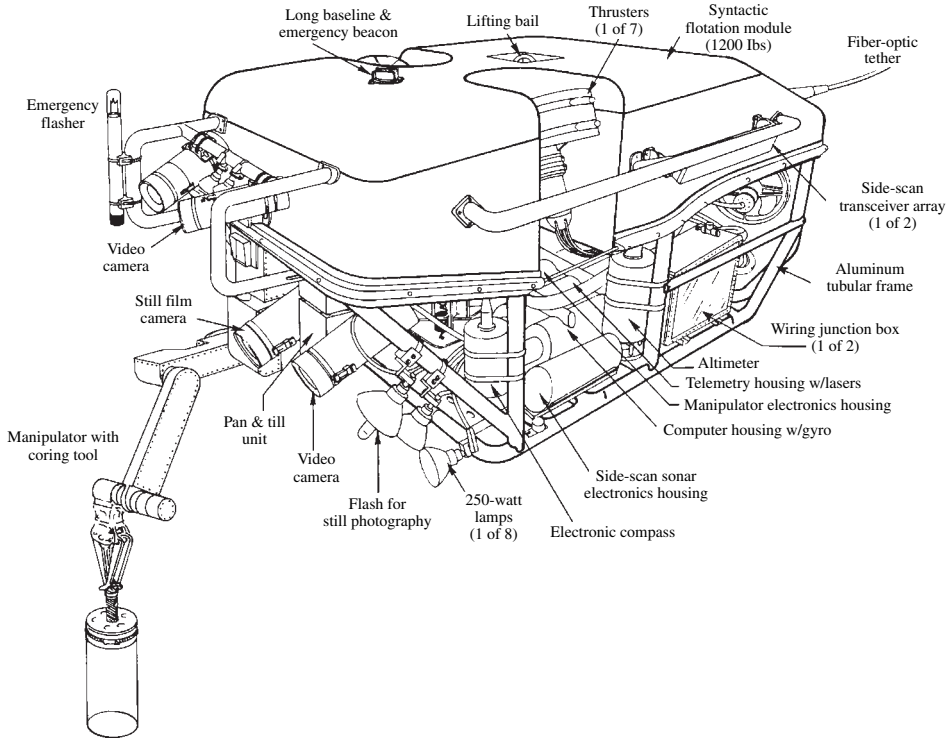


FIGURE 6.9 *Jason* is an underwater, remote-controlled vehicle that has been used to explore the wreckage of the *Lusitania*. The manipulator and cameras comprise some of the vehicle's control systems

Example 6.9

Stability Design via Routh-Hurwitz

PROBLEM: Find the range of gain, K, for the system of Figure 6.10 that will cause the system to be stable, unstable, and marginally stable. Assume $K > 0$.

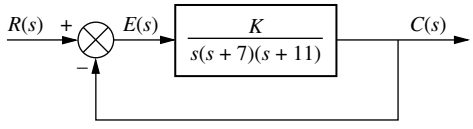


FIGURE 6.10 Feedback control system for Example 6.9

SOLUTION: First find the closed-loop transfer function as

$$T(s) = \frac{K}{s^3 + 18s^2 + 77s + K} \tag{6.20}$$

Next form the Routh table shown as Table 6.15.

TABLE 6.15 Routh table for Example 6.9

s^3	1	77
s^2	18	K
s^1	$\frac{1386 - K}{18}$	
s^0	K	

Since K is assumed positive, we see that all elements in the first column are always positive except the s^1 row. This entry can be positive, zero, or negative, depending upon the value of K. If $K < 1386$, all terms in the first column will be positive, and since there are no sign changes, the system will have three poles in the left half-plane and be *stable*.

If $K > 1386$, the s^1 term in the first column is negative. There are two sign changes, indicating that the system has two right–half-plane poles and one left–half-plane pole, which makes the system *unstable*.

If $K = 1386$, we have an entire row of zeros, which could signify $j\omega$ poles. Returning to the s^2 row and replacing K with 1386, we form the even polynomial

$$P(s) = 18s^2 + 1386 \tag{6.21}$$

Differentiating with respect to s, we have

$$\frac{dP(s)}{ds} = 36s + 0 \tag{6.22}$$

Replacing the row of zeros with the coefficients of Eq. (6.22), we obtain the Routh-Hurwitz table shown as Table 6.16 for the case of $K = 1386$.

TABLE 6.16 Routh table for Example 6.9 with $K = 1386$

s^3	1	77
s^2	18	1386
s^1	~~0~~ 36	
s^0	1386	

Since there are no sign changes from the even polynomial (s^2 row) down to the bottom of the table, the even polynomial has its two roots on the $j\omega$-axis of unit multiplicity. Since there are no sign changes above the even polynomial, the remaining root is in the left half-plane. Therefore the system is *marginally stable.*

MATLAB
ML

Students who are using MATLAB should now run ch6p2 in Appendix B. You will learn how to set up a loop to search for the range of gain to yield stability. This exercise uses MATLAB to do Example 6.9.

Symbolic Math
SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch6sp2 in Appendix F at www.wiley.com/college/nise. You will learn how to use the Symbolic Math Toolbox to calculate the values of cells in a Routh table even if the table contains symbolic objects, such as a variable gain, K. You will see that the Symbolic Math Toolbox and MATLAB yield an alternative way to solve Example 6.9.

The Routh-Hurwitz criterion is often used in limited applications to factor polynomials containing even factors. Let us look at an example.

Example 6.10

Factoring via Routh-Hurwitz

PROBLEM: Factor the polynomial

$$s^4 + 3s^3 + 30s^2 + 30s + 200 \tag{6.23}$$

SOLUTION: Form the Routh table of Table 6.17. We find that the s^1 row is a row of zeros. Now form the even polynomial at the s^2 row:

$$P(s) = s^2 + 10 \tag{6.24}$$

TABLE 6.17 Routh table for Example 6.10

s^4		1	30	200
s^3	~~3~~	1	~~30~~ 10	
s^2	~~20~~	1	~~200~~ 10	
s^1	~~0~~	2	~~0~~ 0	
s^0		10		

This polynomial is differentiated with respect to s in order to complete the Routh table. However, since this polynomial is a factor of the original polynomial in Eq. (6.23), dividing Eq. (6.23) by (6.24) yields $(s^2 + 3s + 20)$ as the other factor. Hence,

$$\begin{aligned} s^4 + 3s^3 + 30s^2 + 30s + 200 &= (s^2 + 10)(s^2 + 3s + 20) \\ &= (s + j3.1623)(s - j3.1623) \\ &\quad \times (s + 1.5 + j4.213)(s + 1.5 - j4.213) \end{aligned} \tag{6.25}$$

Skill-Assessment Exercise 6.3

WileyPLUS
WPCS
Control Solutions

PROBLEM: For a unity feedback system with the forward transfer function

$$G(s) = \frac{K(s+20)}{s(s+2)(s+3)}$$

find the range of K to make the system stable.

ANSWER: $0 < K < 2$

The complete solution is at www.wiley.com/college/nise.

6.5 Stability in State Space

State Space
SS

Up to this point we have examined stability from the s-plane viewpoint. Now we look at stability from the perspective of state space. In Section 4.10, we mentioned that the values of the system's poles are equal to the eigenvalues of the system matrix, $\mathbf{A}$. We stated that the eigenvalues of the matrix $\mathbf{A}$ were solutions of the equation $\det(s\mathbf{I} - \mathbf{A}) = 0$, which also yielded the poles of the transfer function. Eigenvalues appeared again in Section 5.8, where they were formally defined and used to diagonalize a matrix. Let us now formally show that the eigenvalues and the system poles have the same values.

Reviewing Section 5.8, the eigenvalues of a matrix, $\mathbf{A}$, are values of λ that permit a nontrivial solution (other than $\mathbf{0}$) for $\mathbf{x}$ in the equation

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{6.26}$$

In order to solve for the values of λ that do indeed permit a solution for $\mathbf{x}$, we rearrange Eq. (6.26) as follows:

$$\lambda\mathbf{x} - \mathbf{A}\mathbf{x} = \mathbf{0} \tag{6.27}$$

or

$$(\lambda\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0} \tag{6.28}$$

Solving for $\mathbf{x}$ yields

$$\mathbf{x} = (\lambda\mathbf{I} - \mathbf{A})^{-1}\mathbf{0} \tag{6.29}$$

or

$$\mathbf{x} = \frac{\text{adj}(\lambda\mathbf{I} - \mathbf{A})}{\det(\lambda\mathbf{I} - \mathbf{A})}\mathbf{0} \tag{6.30}$$

We see that all solutions will be the null vector except for the occurrence of zero in the denominator. Since this is the only condition where elements of $\mathbf{x}$ will be $0/0$, or indeterminate, it is the only case where a nonzero solution is possible.

The values of λ are calculated by forcing the denominator to zero:

$$\det(\lambda\mathbf{I} - \mathbf{A}) = 0 \tag{6.31}$$

This equation determines the values of λ for which a nonzero solution for $\mathbf{x}$ in Eq. (6.26) exists. In Section 5.8, we defined $\mathbf{x}$ as *eigenvectors* and the values of λ as the *eigenvalues* of the matrix $\mathbf{A}$.

Let us now relate the eigenvalues of the system matrix, $\mathbf{A}$, to the system's poles. In Chapter 3 we derived the equation of the system transfer function, Eq. (3.73), from the state equations. The system transfer function has $\det(s\mathbf{I} - \mathbf{A})$ in the denominator because of the presence of $(s\mathbf{I} - \mathbf{A})^{-1}$. Thus,

$$\det(s\mathbf{I} - \mathbf{A}) = 0 \tag{6.32}$$

is the characteristic equation for the system from which the system poles can be found.

Since Eqs. (6.31) and (6.32) are identical apart from a change in variable name, we conclude that the eigenvalues of the matrix $\mathbf{A}$ are identical to the system's poles before cancellation of common poles and zeroes in the transfer function. Thus, we can determine the stability of a system represented in state space by finding the eigenvalues of the system matrix, $\mathbf{A}$, and determining their locations on the s-plane.

Example 6.11

Stability in State Space

PROBLEM: Given the system

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 3 & 1 \\ 2 & 8 & 1 \\ -10 & -5 & -2 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} u \tag{6.33a}$$

$$y = [1 \quad 0 \quad 0]\mathbf{x} \tag{6.33b}$$

find out how many poles are in the left half-plane, in the right half-plane, and on the $j\omega$-axis.

SOLUTION: First form $(s\mathbf{I} - \mathbf{A})$:

$$(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 3 & 1 \\ 2 & 8 & 1 \\ -10 & -5 & -2 \end{bmatrix} = \begin{bmatrix} s & -3 & -1 \\ -2 & s-8 & -1 \\ 10 & 5 & s+2 \end{bmatrix} \tag{6.34}$$

Now find the $\det(s\mathbf{I} - \mathbf{A})$:

$$\det(s\mathbf{I} - \mathbf{A}) = s^3 - 6s^2 - 7s - 52 \tag{6.35}$$

Using this polynomial, form the Routh table of Table 6.18.

TABLE 6.18 Routh table for Example 6.11

s^3	1	-7
s^2	$\cancel{6}$ -3	$\cancel{52}$ -26
s^1	$\cancel{\frac{47}{3}}$ -1	$\cancel{0}$ 0
s^0	-26	

Since there is one sign change in the first column, the system has one right–half-plane pole and two left–half-plane poles. It is therefore unstable. Yet, you may question the possibility that if a nonminimum-phase zero cancels the unstable pole, the system will be stable. However, in practice, the nonminimum-phase zero or unstable pole will shift due to a slight change in the system's parameters. This change will cause the system to become unstable.

MATLAB
ML

Students who are using MATLAB should now run ch6p3 in Appendix B. You will learn how to determine the stability of a system represented in state space by finding the eigenvalues of the system matrix. This exercise uses MATLAB to do Example 6.11.

Skill-Assessment Exercise 6.4

WileyPLUS
WPCS
Control Solutions

PROBLEM: For the following system represented in state space, find out how many poles are in the left half-plane, in the right half-plane, and on the $j\omega$-axis.

$$\dot{\mathbf{x}} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 7 & 1 \\ -3 & 4 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

$$\mathrm{y} = [0 \quad 1 \quad 0]\mathbf{x}$$

ANSWER: Two rhp and one lhp.

The complete solution is at www.wiley.com/college/nise.

TryIt 6.3

Use the following MATLAB statements to find the eigenvalues of the system described in Skill-Assessment Exercise 6.4.

```
A=[2  1  1
   1  7  1
  -3  4  -5];
Eig=eig(A)
```

In this section, we have evaluated the stability of feedback control systems from the state-space perspective. Since the closed-loop poles and the eigenvalues of a system are the same, the stability requirement of a system represented in state space dictates that the eigenvalues cannot be in the right half of the s-plane or be multiple on the $j\omega$-axis.

We can obtain the eigenvalues from the state equations without first converting to a transfer function to find the poles: The equation $\det(s\mathbf{I} - \mathbf{A}) = 0$ yields the eigenvalues directly. If $\det(s\mathbf{I} - \mathbf{A})$, a polynomial in s, cannot be factored easily, we can apply the Routh-Hurwitz criterion to it to evaluate how many eigenvalues are in each region of the s-plane.

We now summarize this chapter, first with case studies and then with a written summary. Our case studies include the antenna azimuth position control system and the UFSS. Stability is as important to these systems as it is to the system shown in Figure 6.11.



FIGURE 6.11 The FANUC M-410iB^{TM} has 4 axes of motion. It is seen here moving and stacking sacks of chocolate

Case Studies

Antenna Control: Stability Design via Gain

Design
D

This chapter has covered the elements of stability. We saw that stable systems have their closed-loop poles in the left half of the s-plane. As the loop gain is changed, the locations of the poles are also changed, creating the possibility that the poles can move into the right half of the s-plane, which yields instability. Proper gain settings are essential for the stability of closed-loop systems. The following case study demonstrates the proper setting of the loop gain to ensure stability.

PROBLEM: You are given the antenna azimuth position control system shown on the front endpapers, Configuration 1. Find the range of preamplifier gain required to keep the closed-loop system stable.

SOLUTION: The closed-loop transfer function was derived in the case studies in Chapter 5 as

$$T(s) = \frac{6.63K}{s^3 + 101.71s^2 + 171s + 6.63K} \tag{6.36}$$

Using the denominator, create the Routh table shown as Table 6.19. The third row of the table shows that a row of zeros occurs if $K = 2623$. This value of K makes the system marginally stable. Therefore, there will be no sign changes in the first column if $0 < K < 2623$. We conclude that, for stability, $0 < K < 2623$.

TABLE 6.19 Routh table for antenna control case study

s^3	1	171
s^2	101.71	$6.63K$
s^1	$17392.41–6.63K$	0
s^0	$6.63K$	

CHALLENGE: We now give you a problem to test your knowledge of this chapter's objectives. Refer to the antenna azimuth position control system shown on the front endpapers, Configuration 2. Find the range of preamplifier gain required to keep the closed-loop system stable.

UFSS Vehicle: Stability Design via Gain

Design
D

For this case study, we return to the UFSS vehicle and study the stability of the pitch control system, which is used to control depth. Specifically, we find the range of pitch gain that keeps the pitch control loop stable.

PROBLEM: The pitch control loop for the UFSS vehicle (*Johnson*, *1980*) is shown on the back endpapers. Let $K_2 = 1$ and find the range of K_1 that ensures that the closed-loop pitch control system is stable.

SOLUTION: The first step is to reduce the pitch control system to a single, closed-loop transfer function. The equivalent forward transfer function, $G_e(s)$, is

$$G_e(s) = \frac{0.25K_1(s + 0.435)}{s^4 + 3.456s^3 + 3.457s^2 + 0.719s + 0.0416} \tag{6.37}$$

With unity feedback the closed-loop transfer function, $T(s)$, is

$$T(s) = \frac{0.25K_1(s + 0.435)}{s^4 + 3.456s^3 + 3.457s^2 + (0.719 + 0.25K_1)s + (0.0416 + 0.109K_1)} \tag{6.38}$$

The denominator of Eq. (6.38) is now used to form the Routh table shown as Table 6.20.

TABLE 6.20 Routh table for UFSS case study

s^4	1	3.457	$0.0416 + 0.109K_1$
s^3	3.456	$0.719 + 0.25K_1$	
s^2	$11.228 - 0.25K_1$	$0.144 + 0.377K_1$	
s^1	$\dfrac{-0.0625K_1^2 + 1.324K_1 + 7.575}{11.228 - 0.25K_1}$		
s^0	$0.144 + 0.377K_1$		

Note: Some rows have been multiplied by a positive constant for convenience.

Looking at the first column, the s^4 and s^3 rows are positive. Thus, all elements of the first column must be positive for stability. For the first column of the s^2 row to be positive, $-\infty < K_1 < 44.91$. For the first column of the s^1 row to be positive, the numerator must be positive, since the denominator is positive from the previous step. The solution to the quadratic term in the numerator yields roots of $K_1 = -4.685$ and 25.87. Thus, for a positive numerator, $-4.685 < K_1 < 25.87$. Finally, for the first column of the s^0 row to be positive, $-0.382 < K_1 < \infty$. Using all three conditions, stability will be ensured if $-0.382 < K_1 < 25.87$.

CHALLENGE: You are now given a problem to test your knowledge of this chapter's objectives. For the UFSS vehicle (*Johnson*, *1980*) heading control system shown on the back endpapers and introduced in the UFSS case study challenge in Chapter 5, do the following:

MATLAB
ML

a. Find the range of heading gain that ensures the vehicle's stability. Let $K_2 = 1$

b. `Repeat Part` **a** `using MATLAB.`

In our case studies, we calculated the ranges of gain to ensure stability. The student should be aware that although these ranges yield stability, setting gain within these limits may not yield the desired transient response or steady-state error characteristics. In Chapters 9 and 11, we will explore design techniques, other than simple gain adjustment, that yield more flexibility in obtaining desired characteristics.

Summary

In this chapter, we explored the concepts of system stability from both the classical and the state-space viewpoints. We found that for linear systems, *stability* is based on a natural response that decays to zero as time approaches infinity. On the other hand, if the natural response increases without bound, the forced response is overpowered by the natural response, and we lose control. This condition is known as *instability*. A third possibility exists: The natural response may neither decay nor grow without bound but oscillate. In this case, the system is said to be *marginally stable*.

We also used an alternative definition of stability when the natural response is not explicitly available. This definition is based on the total response and says that a system is stable if every bounded input yields a bounded output (BIBO) and unstable if any bounded input yields an unbounded output.

Mathematically, stability for linear, time-invariant systems can be determined from the location of the closed-loop poles:

- If the poles are only in the left half-plane, the system is stable.
- If any poles are in the right half-plane, the system is unstable.
- If the poles are on the $j\omega$-axis and in the left half-plane, the system is marginally stable as long as the poles on the $j\omega$-axis are of unit multiplicity; it is unstable if there are any multiple $j\omega$ poles.

Unfortunately, although the open-loop poles may be known, we found that in higher-order systems it is difficult to find the closed-loop poles without a computer program.

The *Routh-Hurwitz criterion* lets us find how many poles are in each of the sections of the s-plane without giving us the coordinates of the poles. Just knowing that there are poles in the right half-plane is enough to determine that a system is unstable. Under certain limited conditions, when an even polynomial is present, the Routh table can be used to factor the system's characteristic equation.

Obtaining stability from the state-space representation of a system is based on the same concept—the location of the roots of the characteristic equation. These roots are equivalent to the eigenvalues of the system matrix and can be found by solving $\det(s\mathbf{I} - \mathbf{A}) = 0$. Again, the Routh-Hurwitz criterion can be applied to this polynomial. The point is that the state-space representation of a system need not be converted to a transfer function in order to investigate stability. In the next chapter, we will look at steady-state errors, the last of three important control system requirements we emphasize.

Review Questions

1. What part of the output response is responsible for determining the stability of a linear system?
2. What happens to the response named in Question 1 that creates instability?

3. What would happen to a physical system that becomes unstable?
4. Why are marginally stable systems considered unstable under the BIBO definition of stability?
5. Where do system poles have to be to ensure that a system is not unstable?
6. What does the Routh-Hurwitz criterion tell us?
7. Under what conditions would the Routh-Hurwitz criterion easily tell us the actual location of the system's closed-loop poles?
8. What causes a zero to show up only in the first column of the Routh table?
9. What causes an entire row of zeros to show up in the Routh table?
10. Why do we sometimes multiply a row of a Routh table by a positive constant?
11. Why do we not multiply a row of a Routh table by a negative constant?
12. If a Routh table has two sign changes above the even polynomial and five sign changes below the even polynomial, how many right–half-plane poles does the system have?
13. Does the presence of an entire row of zeros always mean that the system has $j\omega$ poles?
14. If a seventh-order system has a row of zeros at the s^3 row and two sign changes below the s^4 row, how many $j\omega$ poles does the system have?

State Space SS

15. Is it true that the eigenvalues of the system matrix are the same as the closed-loop poles?

State Space SS

16. How do we find the eigenvalues?

Problems

1. Tell how many roots of the following polynomial are in the right half-plane, in the left half-plane, and on the $j\omega$-axis: [Section: 6.2]

$$P(s) = s^5 + 3s^4 + 5s^3 + 4s^2 + s + 3$$

2. Tell how many roots of the following polynomial are in the right half-plane, in the left half-plane, and on the $j\omega$-axis: [Section: 6.3]

$$P(s) = s^5 + 6s^3 + 5s^2 + 8s + 20$$

3. Using the Routh table, tell how many poles of the following function are in the right half-plane, in the left half-plane, and on the $j\omega$-axis: [Section: 6.3]

WileyPLUS WPCS Control Solutions

$$T(s) = \frac{s + 8}{s^5 - s^4 + 4s^3 - 4s^2 + 3s - 2}$$

4. The closed-loop transfer function of a system is [Section: 6.3]

$$T(s) = \frac{s^3 + 2s^2 + 7s + 21}{s^5 - 2s^4 + 3s^3 - 6s^2 + 2s - 4}$$

Determine how many closed-loop poles lie in the right half-plane, in the left half-plane, and on the $j\omega$-axis.

5. How many poles are in the right half-plane, in the left half-plane, and on the $j\omega$-axis for the open-loop system of Figure P6.1? [Section: 6.3]

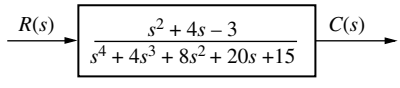


FIGURE P6.1

6. How many poles are in the right half-plane, the left half-plane, and on the $j\omega$-axis for the open-loop system of Figure P6.2? [Section: 6.3]

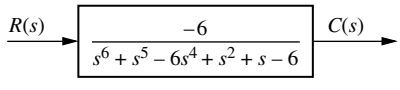


FIGURE P6.2

7. Use MATLAB to find the pole locations for the system of Problem 6.

MATLAB ML

8. Use MATLAB and the Symbolic Math Toolbox to generate a Routh table to solve Problem 3.

Symbolic Math SM

9. Determine whether the unity feedback system of Figure P6.3 is stable if [Section: 6.2]

WileyPLUS WPCS Control Solutions

$$G(s) = \frac{240}{(s+1)(s+2)(s+3)(s+4)}$$

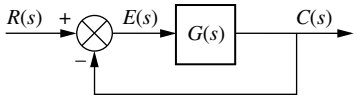


FIGURE P6.3

10. Use MATLAB to find the pole locations for the system of Problem 9.

MATLAB ML

11. Consider the unity feedback system of Figure P6.3 with

$$G(s) = \frac{1}{4s^2(s^2+1)}$$

Using the Routh-Hurwitz criterion, find the region of the s-plane where the poles of the closed-loop system are located. [Section: 6.3]

12. In the system of Figure P6.3, let

$$G(s) = \frac{K(s+2)}{s(s-1)(s+3)}$$

Find the range of K for closed-loop stability. [Section: 6.4]

13. Given the unity feedback system of Figure P6.3 with [Section: 6.3]

$$G(s) = \frac{84}{s(s^7+5s^6+12s^5+25s^4+45s^3+50s^2+82s+60)}$$

tell how many poles of the closed-loop transfer function lie in the right half-plane, in the left half-plane, and on the $j\omega$-axis. [Section: 6.3]

14. Using the Routh-Hurwitz criterion and the unity feedback system of Figure P6.3 with

$$G(s) = \frac{1}{2s^4+5s^3+s^2+2s}$$

tell whether or not the closed-loop system is stable. [Section: 6.2]

15. Given the unity feedback system of Figure P6.3 with

$$G(s) = \frac{8}{s(s^6-2s^5-s^4+2s^3+4s^2-8s-4)}$$

tell how many closed-loop poles are located in the right half-plane, in the left half-plane, and on the $j\omega$-axis. [Section: 6.3]

16. Repeat Problem 15 using MATLAB.

MATLAB ML

17. Consider the following Routh table. Notice that the s^5 row was originally all zeros. Tell how many roots of the original polynomial were in the right half-plane, in the left half-plane, and on the $j\omega$-axis. [Section: 6.3]

s^7	1	2	−1	−2
s^6	1	2	−1	−2
s^5	3	4	−1	0
s^4	1	−1	−3	0
s^3	7	8	0	0
s^2	−15	−21	0	0
s^1	−9	0	0	0
s^0	−21	0	0	0

18. For the system of Figure P6.4, tell how many closed-loop poles are located in the right half-plane, in the left half-plane, and on the $j\omega$-axis. Notice that there is positive feedback. [Section: 6.3]

WileyPLUS WPCS Control Solutions

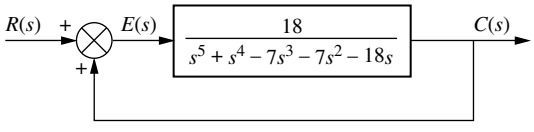


FIGURE P6.4

19. Using the Routh-Hurwitz criterion, tell how many closed-loop poles of the system shown in Figure P6.5 lie in the left half-plane, in the right half-plane, and on the $j\omega$-axis. [Section: 6.3]

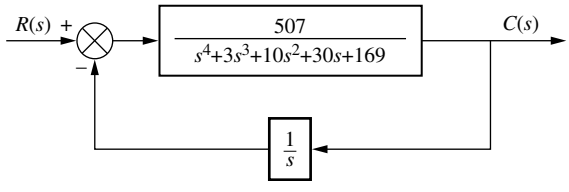


FIGURE P6.5

20. Determine if the unity feedback system of Figure P6.3 with

$$G(s) = \frac{K(s^2+1)}{(s+1)(s+2)}$$

can be unstable. [Section: 6.4]

21. For the unity feedback system of Figure P6.3 with

$$G(s) = \frac{K(s+6)}{s(s+1)(s+4)}$$

determine the range of K to ensure stability. [Section: 6.4]

22. In the system of Figure P6.3, let

$$G(s) = \frac{K(s-a)}{s(s-b)}$$

Find the range of K for closed-loop stability when: [Section: 6.4]

a. $a < 0, \quad b < 0$

b. $a < 0, \quad b > 0$

c. $a > 0, \quad b < 0$

d. $a > 0, \quad b > 0$

WileyPLUS WPCS Control Solutions

23. For the unity feedback system of Figure P6.3 with

$$G(s) = \frac{K(s+3)(s+5)}{(s-2)(s-4)}$$

determine the range of K for stability. [Section: 6.4]

MATLAB ML

24. Repeat Problem 23 using MATLAB.

Symbolic Math SM

25. Use MATLAB and the Symbolic Math Toolbox to generate a Routh table in terms of K to solve Problem 23.

26. Find the range of K for stability for the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K(s+4)(s-4)}{(s^2+3)}$$

27. For the unity feedback system of Figure P6.3 with

$$G(s) = \frac{K(s+1)}{s^4(s+2)}$$

find the range of K for stability. [Section: 6.4]

28. Find the range of gain, K, to ensure stability in the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K(s-2)(s+4)(s+5)}{(s^2+12)}$$

29. Find the range of gain, K, to ensure stability in the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K(s+2)}{(s^2+1)(s+4)(s-1)}$$

30. Using the Routh-Hurwitz criterion, find the value of K that will yield oscillations for the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K}{(s+77)(s+27)(s+38)}$$

31. Use the Routh-Hurwitz criterion to find the range of K for which the system of Figure P6.6 is stable. [Section: 6.4]

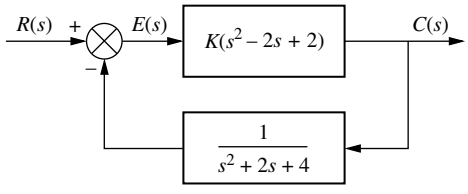


FIGURE P6.6

32. Repeat Problem 31 for the system of Figure P6.7. [Section: 6.4]

WileyPLUS WPCS Control Solutions

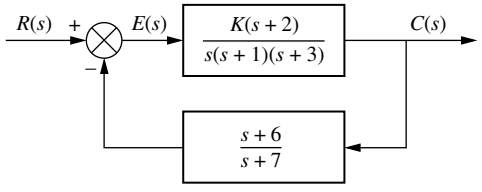


FIGURE P6.7

33. Given the unity feedback system of Figure P6.3 with

$$G(s) = \frac{K(s+4)}{s(s+1.2)(s+2)}$$

find the following: [Section: 6.4]

a. The range of K that keeps the system stable

b. The value of K that makes the system oscillate

c. The frequency of oscillation when K is set to the value that makes the system oscillate

34. Repeat Problem 33 for [Section: 6.4]

$$G(s) = \frac{K(s-1)(s-2)}{(s+2)(s^2+2s+2)}$$

35. For the system shown in Figure P6.8, find the value of gain, K, that will make the system oscillate. Also, find the frequency of oscillation. [Section: 6.4]

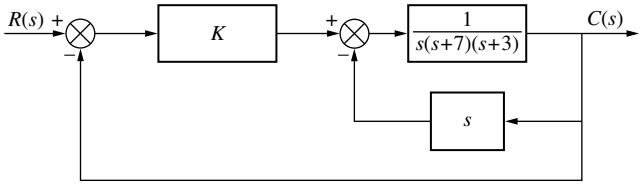


FIGURE P6.8

36. Given the unity feedback system of Figure P6.3 with [Section: 6.4]

WileyPLUS WPCS Control Solutions

$$G(s) = \frac{Ks(s+2)}{(s^2-4s+8)(s+3)}$$

a. Find the range of K for stability.

b. Find the frequency of oscillation when the system is marginally stable.

37. Repeat Problem 36 using MATLAB.

MATLAB ML

38. For the unity feedback system of Figure P6.3 with

$$G(s) = \frac{K(s+2)}{(s^2+1)(s+4)(s-1)}$$

find the range of K for which there will be only two closed-loop, right–half-plane poles. [Section: 6.4]

39. For the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K}{(s+1)^3(s+4)}$$

a. Find the range of K for stability.

b. Find the frequency of oscillation when the system is marginally stable.

40. Given the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K}{(s+49)(s^2+4s+5)}$$

a. Find the range of K for stability.

b. Find the frequency of oscillation when the system is marginally stable.

41. Using the Routh-Hurwitz criterion and the unity feedback system of Figure P6.3 with [Section: 6.4]

$$G(s) = \frac{K}{s(s+1)(s+2)(s+5)}$$

a. Find the range of K for stability.

b. Find the value of K for marginal stability.

c. Find the actual location of the closed-loop poles when the system is marginally stable.

42. Find the range of K to keep the system shown in Figure P6.9 stable. [Section: 6.4]

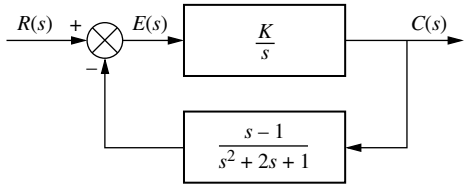


FIGURE P6.9

43. Find the value of K in the system of Figure P6.10 that will place the closed-loop poles as shown. [Section: 6.4]

WileyPLUS WPCS Control Solutions

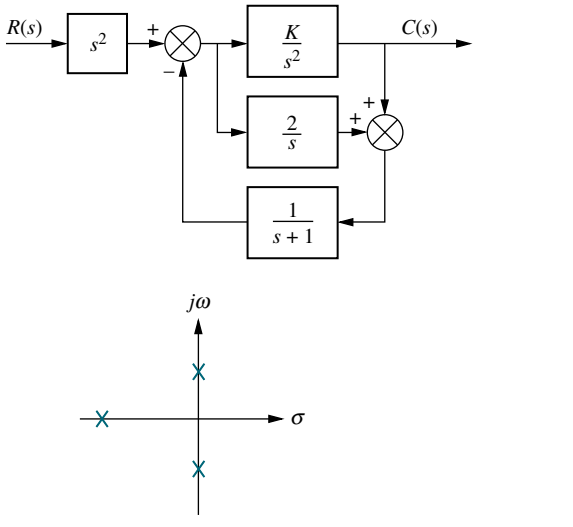


FIGURE P6.10 Closed-loop system with pole plot

44. The closed-loop transfer function of a system is

$$T(s) = \frac{s^2 + K_1 s + K_2}{s^4 + K_1 s^3 + K_2 s^2 + 5s + 1}$$

Determine the range of K_1 in order for the system to be stable. What is the relationship between K_1 and K_2 for stability? [Section: 6.4]

45. For the transfer function below, find the constraints on K_1 and K_2 such that the function will have only two $j\omega$ poles. [Section: 6.4]

$$T(s) = \frac{K_1 s + K_2}{s^4 + K_1 s^3 + s^2 + K_2 s + 1}$$

46. The transfer function relating the output engine fan speed (rpm) to the input main burner fuel flow rate (lb/h) in a short takeoff and landing (STOL) fighter aircraft, ignoring the coupling between engine fan speed and the pitch control command, is (*Schierman, 1992*) [Section: 6.4]

$$G(s) = \frac{1.3s^7 + 90.5s^6 + 1970s^5 + 15,000s^4 + 3120s^3 - 41,300s^2 - 5000s - 1840}{s^8 + 103s^7 + 1180s^6 + 4040s^5 + 2150s^4 - 8960s^3 - 10,600s^2 - 1550s - 415}$$

a. Find how many poles are in the right half-plane, in the left half-plane, and on the $j\omega$-axis.

b. Is this open-loop system stable?

47. An interval polynomial is of the form

$$P(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4 + a_5 s^5 + \cdots$$

with its coefficients belonging to intervals $x_i \le a_i \le y_i$, where x_i, y_i are prescribed constants. Kharitonov's theorem says that an interval polynomial has all its roots in the left half-plane if each one of the following four polynomials has its roots in the left half-plane (*Minichelli, 1989*):

$$K_1(s) = x_0 + x_1 s + y_2 s^2 + y_3 s^3 + x_4 s^4 + x_5 s^5 + y_6 s^6 + \cdots$$
$$K_2(s) = x_0 + y_1 s + y_2 s^2 + x_3 s^3 + x_4 s^4 + y_5 s^5 + y_6 s^6 + \cdots$$
$$K_3(s) = y_0 + x_1 s + x_2 s^2 + y_3 s^3 + y_4 s^4 + x_5 s^5 + x_6 s^6 + \cdots$$
$$K_4(s) = y_0 + y_1 s + x_2 s^2 + x_3 s^3 + y_4 s^4 + y_5 s^5 + x_6 s^6 + \cdots$$

Use Kharitonov's theorem and the Routh-Hurwitz criterion to find if the following polynomial has any zeros in the right–half-plane.

$$P(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3$$
$$2 \le a_0 \le 4; \quad 1 \le a_1 \le 2; \quad 4 \le a_2 \le 6; \quad a_3 = 1$$

48. A linearized model of a torque-controlled crane hoisting a load with a fixed rope length is

$$P(s) = \frac{X_T(s)}{F_T(s)} = \frac{1}{m_T} \frac{s^2 + \omega_0^2}{s^2(s^2 + a\omega_0^2)}$$

where $\omega_0 = \sqrt{\frac{g}{L}}$, L = the rope length, m_T = the mass of the car, a = the combined rope and car mass, f_T = the force input applied to the car, and x_T = the resulting rope displacement (*Marttinen, 1990*). If the system is controlled in a feedback configuration by placing it in a loop as shown in Figure P6.11, with $K > 0$, where will the closed-loop poles be located?

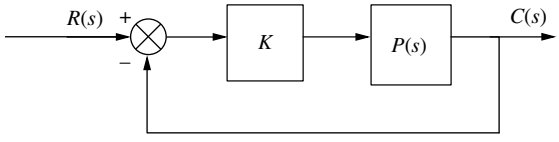


FIGURE P6.11

49. The read/write head assembly arm of a computer hard disk drive (HDD) can be modeled as a rigid rotating body with inertia I_b. Its dynamics can be described with the transfer function

$$P(s) = \frac{X(s)}{F(s)} = \frac{1}{I_b s^2}$$

where $X(s)$ is the displacement of the read/write head and $F(s)$ is the applied force (*Yan, 2003*). Show that if the HDD is controlled in the configuration shown in Figure P6.11, the arm will oscillate and cannot be positioned with any precision over a HDD track. Find the oscillation frequency.

50. A system is represented in state space as State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 3 \\ 2 & 2 & -4 \\ 1 & -4 & 3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u$$
$$y = [1 \quad 1 \quad 0]\mathbf{x}$$

Determine how many eigenvalues are in the right half-plane, in the left half-plane, and on the $j\omega$-axis. [Section: 6.5]

51. Use MATLAB to find the eigenvalues of the following system:

MATLAB ML State Space SS

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & -4 \\ -1 & 1 & 8 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [0 \quad 0 \quad 1]\mathbf{x}$$

52. The following system in state space represents the forward path of a unity feedback system. Use the Routh-Hurwitz criterion to determine if the closed-loop system is stable. [Section: 6.5]

State Space SS WileyPLUS WPCS Control Solutions

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 3 \\ -3 & -4 & -5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [0 \quad 1 \quad 1]\mathbf{x}$$

53. Repeat Problem 52 using MATLAB.

MATLAB ML

54. A Butterworth polynomial is of the form

$$B_n(s) = 1 + (-1)^n \left(\frac{s}{\omega_c}\right)^{2n}, \quad n > 0.$$

Use the Routh-Hurwitz criteria to find the zeros of a Butterworth polynomial for:

a. $n = 1$;

b. $n = 2$

DESIGN PROBLEMS

55. A model for an airplane's pitch loop is shown in Figure P6.12. Find the range of gain, K, that will keep the system stable. Can the system ever be unstable for positive values of K?

WileyPLUS WPCS Control Solutions

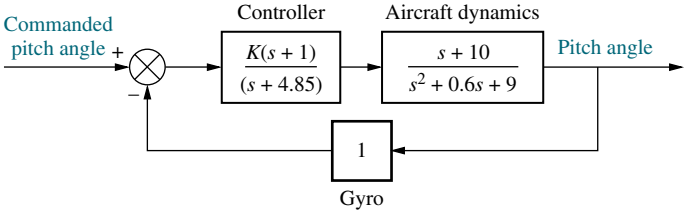


FIGURE P6.12 Aircraft pitch loop model

56. A common application of control systems is in regulating the temperature of a chemical process (Figure P6.13). The flow of a chemical reactant to a process is controlled by an actuator and valve. The reactant causes the temperature in the vat to change. This temperature is sensed and compared to a desired set-point temperature in a closed loop, where the flow of reactant is adjusted to yield the desired temperature. In Chapter 9, we will learn how a PID controller is used to improve the performance of such process control systems. Figure P6.13 shows the control system prior to the addition of the PID controller. The PID controller is replaced by the shaded box with a gain of unity. For this system, prior to the design of the PID controller, find the range of amplifier gain, K, to keep the system stable.

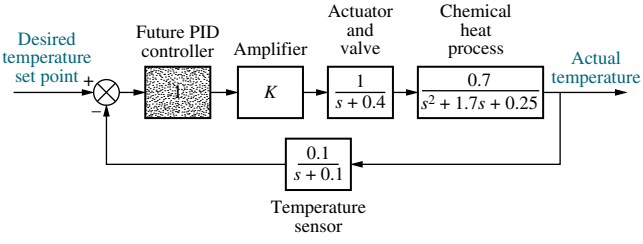
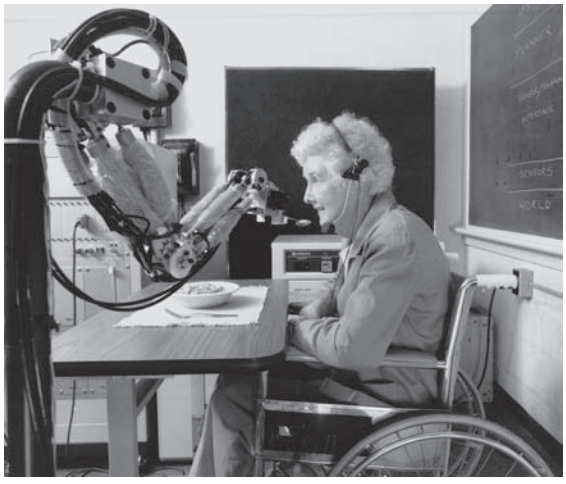
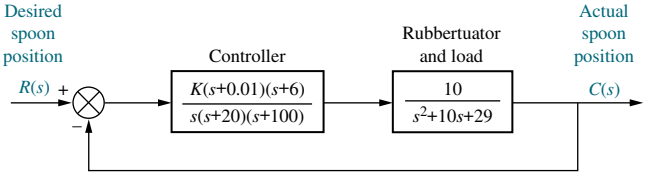


FIGURE P6.13 Block diagram of a chemical process control system

57. A robot arm called ISAC (Intelligent Soft Arm Control) can be used as part of a system to feed people with disabilities (see Figure P6.14(*a*)). The control system guides the spoon to the food and then to a position near the person's mouth. The arm uses a special pneumatically controlled actuator called a rubbertuator. Rubbertuators consist of rubber tubes covered with fiber cord. The actuator contracts in length when pneumatic pressure is increased and expands in length when pressure is decreased. This expansion and contraction in length can drive a pulley or other device. A video camera provides the sight for the robot and the tracking loop (*Kara, 1992*). Assume the simplified block diagram shown in Figure P6.14(*b*) for regulating the spoon at a distance from the mouth. Find the range of K for stability. (Use of a program with symbolic capability is recommended.)



(*a*)



(*b*)

FIGURE P6.14 **a.** ISAC used for feeding (Courtesy of Kazuhiko Kawamura, Vanderbilt University.) **b.** simplified block diagram

58. Often an aircraft is required to tow another vehicle, such as a practice target or glider. To stabilize the towed vehicle and prevent it from rolling, pitching, and yawing, an autopilot is built into the towed vehicle. Assume the block diagram shown in Figure P6.15 represents the autopilot roll control system (*Cochran*, *1992*). Find the range of K to keep the roll angle stable.

WileyPLUS WPCS Control Solutions

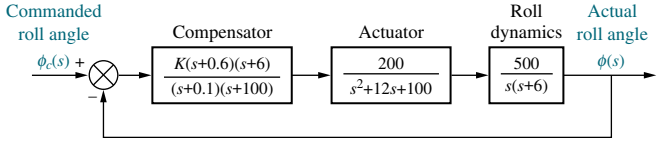


FIGURE P6.15 Towed vehicle roll control

59. Cutting forces should be kept constant during machining operations to prevent changes in spindle speeds or work position. Such changes would deteriorate the accuracy of the work's dimensions. A control system is proposed to control the cutting force. The plant is difficult to model, since the factors that affect cutting force are time varying and not easily predicted. However, assuming the simplified force control model shown in Figure P6.16, use the Routh-Hurwitz criterion to find the range of K to keep the system stable (*Rober*, *1997*).

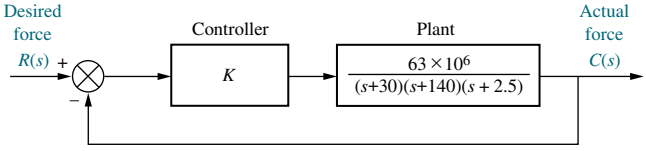
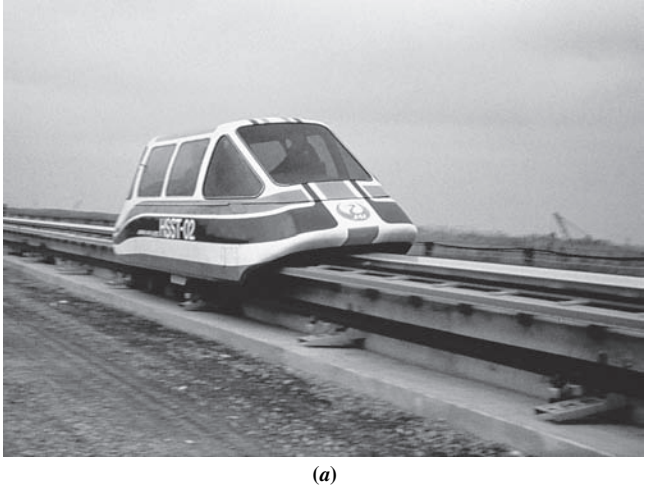
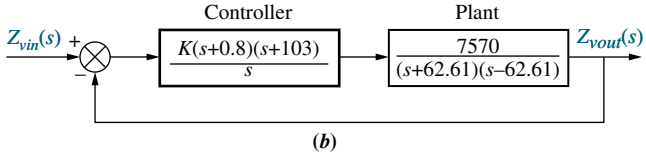


FIGURE P6.16 Cutting force control system (Reprinted with permission of ASME.)

60. Transportation systems that use magnetic levitation can reach very high speeds, since contact friction at the rails is eliminated (see Figure P6.17(*a*)). Electromagnets can produce the force to elevate the vehicle. Figure P6.17(*b*) is a simulation model of a control system that can be used to regulate the magnetic gap. In the figure, $Z_{vin}(s)$ represents a voltage proportional to the desired amount of levitation, or gap. $Z_{vout}(s)$ represents a voltage proportional to the actual amount of levitation. The plant models the dynamic response of the vehicle to



(*a*)



(*b*)

FIGURE P6.17 **a.** A magnetic levitation transportation system (© Japan Air Lines/Photo Researchers); **b.** simplified block diagram (© 1998 IEEE)

signals from the controller (*Bittar*, *1998*). Use the Routh-Hurwitz criterion to find the range of gain, K, to keep the closed loop system stable.

61. A transfer function from indoor radiator power, $\dot{Q}(s)$, to room temperature, $T(s)$, in an 11 m^2 room is

$$P(s) = \frac{T(s)}{\dot{Q}(s)} = \frac{1 \times 10^{-6}s^2 + 1.314 \times 10^{-9}s + 2.66 \times 10^{-13}}{s^3 + 0.00163s^2 + 5.272 \times 10^{-7}s + 3.538 \times 10^{-11}}$$

where $\dot{Q}$ is in watts and T is in °C. (*Thomas*, *2005*). The room's temperature will be controlled by embedding it in a closed loop, such as that of Figure P6.11. Find the range of K for closed-loop stability.

62. During vertical spindle surface grinding, adjustments are made on a multi-axis computer numerical control (CNC) machine by measuring the applied force with a dynamometer and applying appropriate corrections. This feedback force control results in higher homogeneity and better tolerances in the resulting finished product. In a specific experiment with an extremely high feed rate, the transfer function from the desired depth of cut (DOC) to applied force was

$$\frac{F(s)}{DOC(s)} = \frac{K_C}{1 + \dfrac{K_C}{ms^2 + bs + k} - \dfrac{K_C}{K_f}\dfrac{1}{Ts + 1}}$$

where $k = 2.1 \times 10^4$ N/m, $b = 0.78$ Ns/m, $m = 1.2 \times 10^{-4}$ Kg, $K_C = 1.5 \times 10^4$ N/mm and $T = 0.004$ s. K_f is a parameter that is varied to adjust the system. Find the range of K_f under which the system is stable (*Hekman*, *1999*).

63. Figure P6.18 depicts the schematic diagram of a phase shift oscillator.

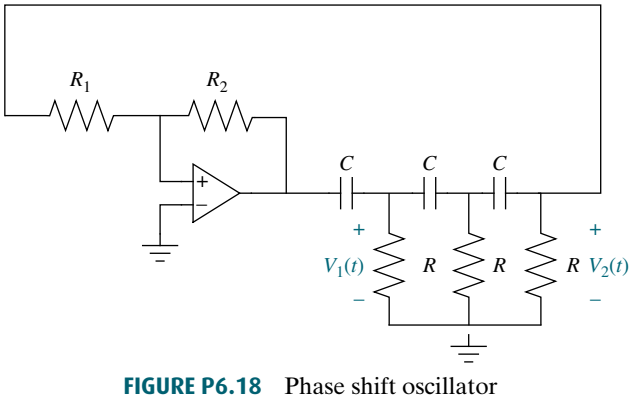


FIGURE P6.18 Phase shift oscillator

The circuit will oscillate if it is designed to have poles on the $j\omega$-axis.

a. Show that the transfer function for the passive network in the circuit is given by

$$\frac{V_2(s)}{V_1(s)} = \frac{-1}{\left(1 + \dfrac{1}{sRC}\right)\left(2 + \dfrac{1}{sRC}\right)^2 - 3 - \dfrac{2}{sRC}}$$

b. Show that the oscillator's characteristic equation is given by

$$1 - K\frac{1}{\left(1 + \dfrac{1}{sRC}\right)\left(2 + \dfrac{1}{sRC}\right)^2 - 3 - \dfrac{2}{sRC}} = 0,$$

where $K = \dfrac{R_2}{R_1}$

c. Use the Routh-Hurwitz criterion to obtain the oscillation condition and the oscillation frequency.

64. In order to obtain a low-cost lithium-ion battery charger, the feedback loop of Figure P6.3 is used, where $G(s) = G_c(s)P(s)$. The following transfer functions have been derived for $G(s)$ (*Tsang*, *2009*):

$$P(s) = \frac{R_1R_2C_1C_2s^2 + (R_1C_1 + R_2C_1 + R_2C_2)s + 1}{C_1(1 + R_2C_2)s}$$

$$G_c(s) = K_p + \frac{K_I}{s}$$

If $R_1 = 0.15\,\Omega$; $R_2 = 0.44\,\Omega$; $C_1 = 7200$ F; and $C_2 = 170$ F use the Routh-Hurwitz criteria to find the range of positive K_P and K_I for which the system is closed-loop stable.

65. Figure P6.19 is a simplified and linearized block diagram of a cascade control system, which is used to control water level in a steam generator of a nuclear power plant (*Wang*, *2009,*).

In this system, the level controller, $G_{LC}(s)$, is the *master* controller and the feed-water flow controller, $G_{FC}(s)$ is the *slave* controller. Using mass balance equations, the water level would ordinarily be regarded as a simple integration process of water flow. In a steam generator, however, steam flow rate and the cooling effect of feed-water change the dynamics of that process. Taking the latter into account and ignoring the much-less pronounced impact of changes in steam flow rate, a first-order lag plus time delay is introduced into the transfer

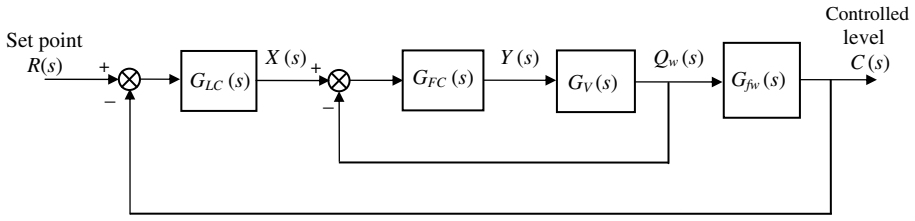


FIGURE P6.19

function, $G_{fw}(s)$, relating the controlled level, $C(s)$, to feed-water flow rate, $Q_w(s)$ as follows:

$$G_{fw}(s) = \frac{C(s)}{Q_w(s)} = \frac{K_1 e^{-\tau_1 S}}{s(T_1 s + 1)} = \frac{2e^{-2S}}{s(25s+1)} \approx \frac{2}{s(25s+1)(2s^2+2s+1)}$$

where $K_1 = 2$ is the process gain, $\tau_1 = 2$ is the pure time delay, and $T_1 = 25$ is the steam generator's time constant. (The expression $e^{-\tau_1 s}$ represents a time delay. This function can be represented by what is known as a *Pade approximation*. This approximation can take on many increasingly complicated forms, depending upon the degree of accuracy required. Here we use the Pade approximation, $e^{-x} \approx \dfrac{1}{1 + x + \dfrac{x^2}{2!}}$, and specific numerical values for the considered steam generator.)

The dynamic characteristics of the control valve are approximated by the transfer function: $G_v(s) = \dfrac{Q_w(s)}{Y(s)} = \dfrac{K_v}{T_v s + 1} = \dfrac{1}{3s+1}$, where K_v is the valve gain and T_v is its time constant.

Given that: $G_{FC}(s) = K_{P_{FC}} + K_{D_{FC}} s = 0.5 + 2s$ and $G_{LC}(s) = K_{P_{LC}} + K_{D_{LC}} s = 0.5 + K s$, use the Routh-Hurwitz criterion to find the range of the level controller's derivative gain, $K_{D_{LC}} = K > 0$, that will keep the system stable.

66. Look-ahead information can be used to automatically steer a bicycle in a closed-loop configuration. A line is drawn in the middle of the lane to be followed, and an arbitrary point is chosen in the vehicle's longitudinal axis. A look-ahead offset is calculated by measuring the distance between the look-ahead point and the reference line and is used by the system to correct the vehicle's trajectory. A linearized model of a particular bicycle traveling on a straight-line path at a fixed longitudinal speed is

$$\begin{bmatrix} \dot{V} \\ \dot{r} \\ \dot{\psi} \\ \dot{Y}_g \end{bmatrix} = \begin{bmatrix} -11.7 & 6.8 & 61.6K & 7.7K \\ -3.5 & -24 & -66.9K & 8.4K \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -10 & 0 \end{bmatrix} \begin{bmatrix} V \\ r \\ \psi \\ Y_g \end{bmatrix}$$

In this model, V = bicycle's lateral velocity, r = bicycle's yaw velocity, ψ = bicycle's yaw acceleration, and Y_g = bicycle's center of gravity coordinate on the y-axis. K is a controller parameter to be chosen by the designer (*Özgüner, 1995*). Use the Routh-Hurwitz citerion to find the range of K for which the system is closed-loop stable.

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

67. **High-speed rail pantograph.** Problem 21 in Chapter 1 discusses active control of a pantograph mechanism for high-speed rail systems. In Problem 79(a), Chapter 5, you found the block diagram for the active pantograph control system. Using your solution for Problem 79(a) in Chapter 5 and the Routh-Hurwitz criterion, find the range of controller gain, K, that will keep the system stable (*O'Connor, 1997*).

68. **Control of HIV/AIDS.** The HIV infection linearized model developed in Problem 82, Chapter 4, can be shown to have the transfer function

$$P(s) = \frac{Y(s)}{U_1(s)} = \frac{-520s - 10.3844}{s^3 + 2.6817s^2 + 0.11s + 0.0126}$$

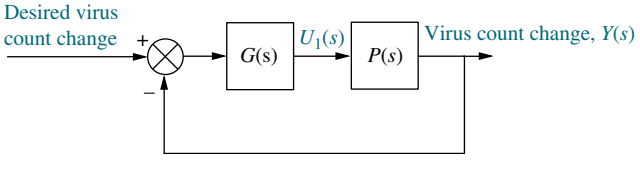


FIGURE P6.20

It is desired to develop a policy for drug delivery to maintain the virus count at prescribed levels. For the purpose of obtaining an appropriate $u_1(t)$, feedback will be used as shown in Figure P6.20 (*Craig, 2004*).

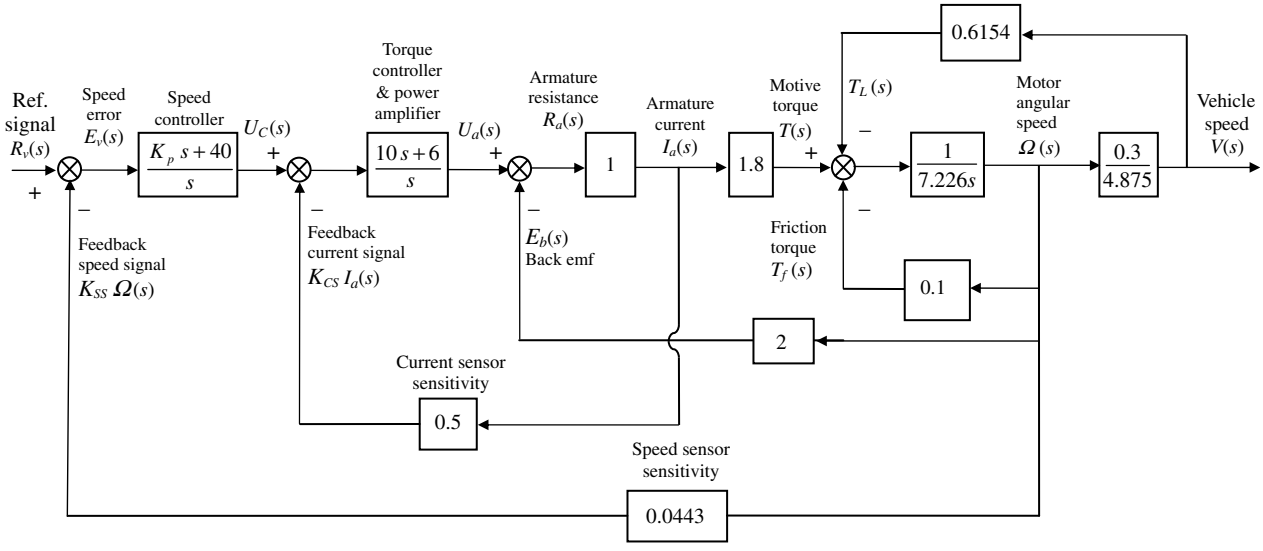


FIGURE P6.21

As a first approach, consider $G(s) = K$, a constant to be selected. Use the Routh-Hurwitz criteria to find the range of K for which the system is closed-loop stable.

69. Hybrid vehicle. Figure P6.21 shows the HEV system presented in Chapter 5, where parameter values have been substituted. It is assumed here that the speed controller has a proportional gain, K_p, to be adjusted. Use the Routh-Hurwitz stability method to find the range of positive K_p for which the system is closed-loop stable (*Graebe, 1995*).

Cyber Exploration Laboratory

Experiment 6.1

Objectives To verify the effect of pole location upon stability. To verify the effect upon stability of loop gain in a negative feedback system.

Minimum Required Software Packages MATLAB, Simulink, and the Control System Toolbox

Prelab

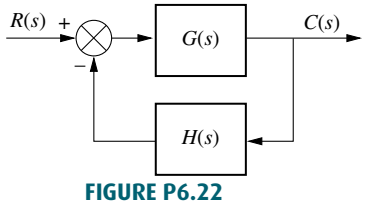


FIGURE P6.22

1. Find the equivalent transfer function of the negative feedback system of Figure P6.22 if

$$G(s) = \frac{K}{s(s+2)^2} \text{ and } H(s) = 1$$

2. For the system of Prelab 1, find two values of gain that will yield closed-loop, overdamped, second-order poles. Repeat for underdamped poles.

3. For the system of Prelab 1, find the value of gain, K, that will make the system critically damped.

4. For the system of Prelab 1, find the value of gain, K, that will make the system marginally stable. Also, find the frequency of oscillation at that value of K that makes the system marginally stable.
5. For each of Prelab 2 through 4, plot on one graph the pole locations for each case and write the corresponding value of gain, K, at each pole.

Lab

1. Using Simulink, set up the negative feedback system of Prelab 1. Plot the step response of the system at each value of gain calculated to yield overdamped, underdamped, critically damped, and marginally stable responses.
2. Plot the step responses for two values of gain, K, above that calculated to yield marginal stability.
3. At the output of the negative feedback system, cascade the transfer function

$$G_1(s) = \frac{1}{s^2 + 4}$$

Set the gain, K, at a value below that calculated for marginal stability and plot the step response. Repeat for K calculated to yield marginal stability.

Postlab

1. From your plots, discuss the conditions that lead to unstable responses.
2. Discuss the effect of gain upon the nature of the step response of a closed-loop system.

Experiment 6.2

Objective To use the LabVIEW Control Design and Simulation Module for stability analysis.

Minimum Required Software Package LabVIEW with the Control Design and Simulation Module

Prelab Select six transfer functions of various orders and use Routh-Hurwitz to determine their stability.

Lab Create a LabVIEW VI that receives the order and the coefficients of the characteristic equation and outputs the location of the poles and information regarding stability.

Postlab Verify the stability of the systems from your Prelab.

Bibliography

Ballard, R. D. The Riddle of the *Lusitania*. *National Geographic*, April 1994, National Geographic Society, Washington, D.C., 1994, pp. 68–85.

Bittar, A., and Sales, R. M. H_2 and H_2 Control for MagLev Vehicles. *IEEE Control Systems*, vol. 18, no. 4, August 1998, pp. 18–25.

Cochran, J. E., Innocenti, M., No, T. S., and Thukral, A. Dynamics and Control of Maneuverable Towed Flight Vehicles. *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, September–October 1992, pp. 1245–1252.

Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria, *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

D'Azzo, J., and Houpis, C. H. *Linear Control System Analysis and Design*, 3d ed. McGraw-Hill, New York, 1988.

Dorf, R. C. *Modern Control Systems*, 5th ed. Addison-Wesley, Reading, MA, 1989.

Graebe, S. F., Goodwin, G. C., and Elsley, G. Control Design and Implementation in Continuous Steel Casting. *IEEE Control Systems, August 1995*, pp. 64–71.

Hekman, K. A., and Liang, S. Y. Compliance Feedback Control for Part Parallelism in Grinding. *International Journal of Manufacturing Technology*, vol. 15, 1999, pp. 64–69.

Hostetter, G. H., Savant, C. J., Jr., and Stefani, R. T. *Design of Feedback Control Systems*, 2d ed. Saunders College Publishing, New York, 1989.

Johnson, H., et al. *Unmanned Free-Swimming Submersible (UFSS) System Description*. NRL Memorandum Report 4393. Naval Research Laboratory, Washington, D.C., 1980.

Kara, A., Kawamura, K., Bagchi, S., and El-Gamal, M. Reflex Control of a Robotic Aid System to Assist the Physically Disabled. *IEEE Control Systems*, June 1992, pp. 71–77.

Martinnen, A., Virkkunen, J., and Salminen, R. T. Control Study with Pilot Crane. *IEEE Transactions on Education*, vol. 33, no. 3, August 1990, pp. 298–305.

Minnichelli, R. J., Anagnost, J. J., and Desoer, C. A. An Elementary Proof of Kharitonov's Stability Theorem with Extensions. *IEEE Transactions on Automatic Control*, vol. 34, 1989, pp. 995–998.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. 1–4.

Özgüner, Ű., Űnyelioglu, K. A., and Haptipoĝlu, C. An Analytical Study of Vehicle Steering Control. Proceedings of the 4th IEEE Conference Control Applications, 1995, pp. 125–130.

Phillips, C. L., and Harbor, R. D. *Feedback Control Systems*, 2d ed. Prentice Hall, Upper Saddle River, NJ, 1991.

Preitl, Z., Bauer, P., and J. Bokor, J. A Simple Control Solution for Traction Motor Used in Hybrid Vehicles. *4th International Symposium on Applied Computational Intelligence and Informatics.* IEEE. 2007.

Rober, S. J., Shin, Y. C., and Nwokah, O. D. I. A Digital Robust Controller for Cutting Force Control in the End Milling Process. *Journal of Dynamic Systems, Measurement, and Control*, vol. 119, June 1997, pp. 146–152.

Routh, E. J. *Dynamics of a System of Rigid Bodies*, 6th ed. Macmillan, London, 1905.

Schierman, J. D., and Schmidt, D. K. Analysis of Airframe and Engine Control Interactions and Integrated Flight/Propulsion Control. *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 6, November–December 1992, pp. 1388–1396.

Thomas, B., Soleimani-Mosheni, M., and Fahlén, P., Feed-Forward in Temperature Control of Buildings. *Energy and Buildings*, vol. 37, 2005, pp. 755–761.

Timothy, L. K., and Bona, B. E. *State Space Analysis: An Introduction*. McGraw-Hill, New York, 1968.

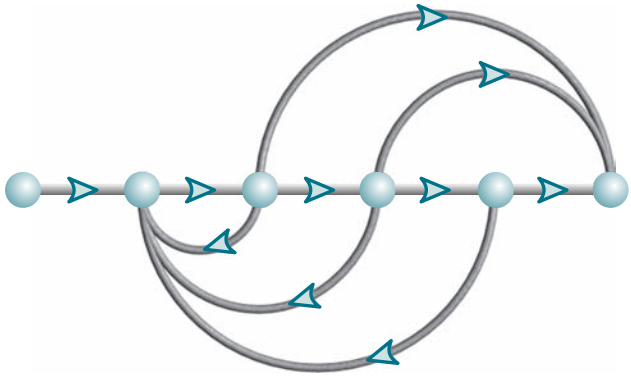
Tsang, K. M., Chan, W. L. A Simple and Low-cost Charger for Lithium-Ion Batteries. *Journal of Power Sources*, vol. 191, 2009, pp. 633–635.

Wang, X.-K., Yang, X.-H., Liu, G., and Qian, H. Adaptive Neuro-Fuzzy Inference System PID controller for steam generator water level of nuclear power plant, Procedings of the Eighth International Conference on Machine Learning and Cybernetics, 2009, pp. 567–572.

Yan, T., and Lin, R. Experimental Modeling and Compensation of Pivot Nonlinearly in Hard Disk Drives. *IEEE Transactions on Magnetics*, vol. 39, 2003, pp. 1064–1069.

Steady-State Errors

7



Chapter Learning Outcomes

After completing this chapter the student will be able to:

- Find the steady-state error for a unity feedback system (Sections 7.1–7.2)
- Specify a system's steady-state error performance (Section 7.3)
- Design the gain of a closed-loop system to meet a steady-state error specification (Section 7.4)
- Find the steady-state error for disturbance inputs (Section 7.5)
- Find the steady-state error for nonunity feedback systems (Section 7.6)
- Find the steady-state error sensitivity to parameter changes (Section 7.7)
- Find the steady-state error for systems represented in state space (Section 7.8)

Case Study Learning Outcomes

You will be able to demonstrate your knowledge of the chapter objectives with case studies as follows:

- Given the antenna azimuth position control system shown on the front endpapers, you will be able to find the preamplifier gain to meet steady-state error performance specifications.
- Given a video laser disc recorder, you will be able to find the gain required to permit the system to record on a warped disc.

7.1 Introduction

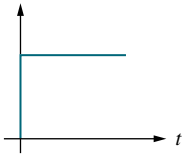
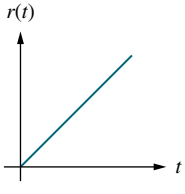
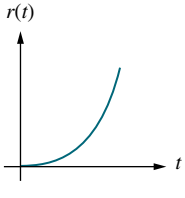
In Chapter 1, we saw that control systems analysis and design focus on three specifications: (1) transient response, (2) stability, and (3) steady-state errors, taking into account the robustness of the design along with economic and social considerations. Elements of transient analysis were derived in Chapter 4 for first- and second-order systems. These concepts are revisited in Chapter 8, where they are extended to higher-order systems. Stability was covered in Chapter 6, where we saw that forced responses were overpowered by natural responses that increase without bound if the system is unstable. Now we are ready to examine steady-state errors. We define the errors and derive methods of controlling them. As we progress, we find that control system design entails trade-offs between desired transient response, steady-state error, and the requirement that the system be stable.

Definition and Test Inputs

Steady-state error is the difference between the input and the output for a prescribed test input as $t \to \infty$. Test inputs used for steady-state error analysis and design are summarized in Table 7.1.

In order to explain how these test signals are used, let us assume a position control system, where the output position follows the input commanded position. Step inputs represent constant position and thus are useful in determining the ability of the control system to position itself with respect to a stationary target, such as a satellite in geostationary orbit (see Figure 7.1). An antenna position control is an example of a system that can be tested for accuracy using step inputs.

TABLE 7.1 Test waveforms for evaluating steady-state errors of position control systems

Waveform	Name	Physical interpretation	Time function	Laplace transform
	Step	Constant position	1	$\frac{1}{s}$
	Ramp	Constant velocity	t	$\frac{1}{s^2}$
	Parabola	Constant acceleration	$\frac{1}{2}t^2$	$\frac{1}{s^3}$

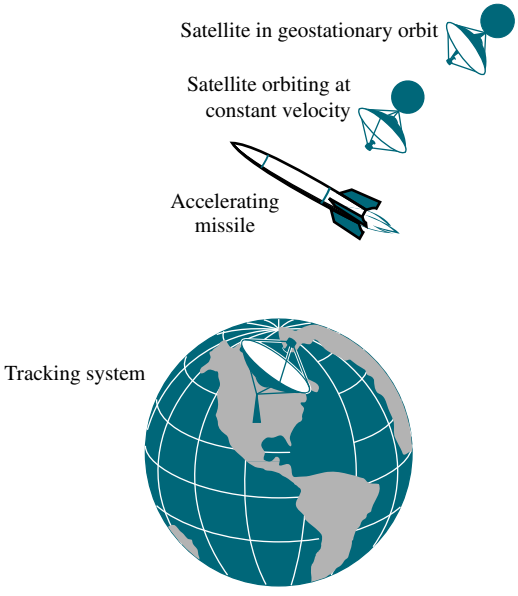


FIGURE 7.1 Test inputs for steady-state error analysis and design vary with target type

Ramp inputs represent constant-velocity inputs to a position control system by their linearly increasing amplitude. These waveforms can be used to test a system's ability to follow a linearly increasing input or, equivalently, to track a constant-velocity target. For example, a position control system that tracks a satellite that moves across the sky at a constant angular velocity, as shown in Figure 7.1, would be tested with a ramp input to evaluate the steady-state error between the satellite's angular position and that of the control system.

Finally, parabolas, whose second derivatives are constant, represent constant-acceleration inputs to position control systems and can be used to represent accelerating targets, such as the missile in Figure 7.1, to determine the steady-state error performance.

Application to Stable Systems

Since we are concerned with the difference between the input and the output of a feedback control system after the steady state has been reached, our discussion is limited to stable systems, where the natural response approaches zero as $t \to \infty$. Unstable systems represent loss of control in the steady state and are not acceptable for use at all. The expressions we derive to calculate the steady-state error can be applied erroneously to an unstable system. Thus, the engineer must check the system for stability while performing steady-state error analysis and design. However, in order to focus on the topic, we assume that all the systems in examples and problems in this chapter are stable. For practice, you may want to test some of the systems for stability.

Evaluating Steady-State Errors

Let us examine the concept of steady-state errors. In Figure 7.2(a) a step input and two possible outputs are shown. Output 1 has zero steady-state error, and output 2 has a finite steady-state error, $e_2(\infty)$. A similar example is shown in Figure 7.2(b), where a ramp input is compared with output 1, which has zero steady-state error, and output 2, which has a finite steady-state error, $e_2(\infty)$, as measured vertically between the input and output 2 after the transients have died down. For the ramp input

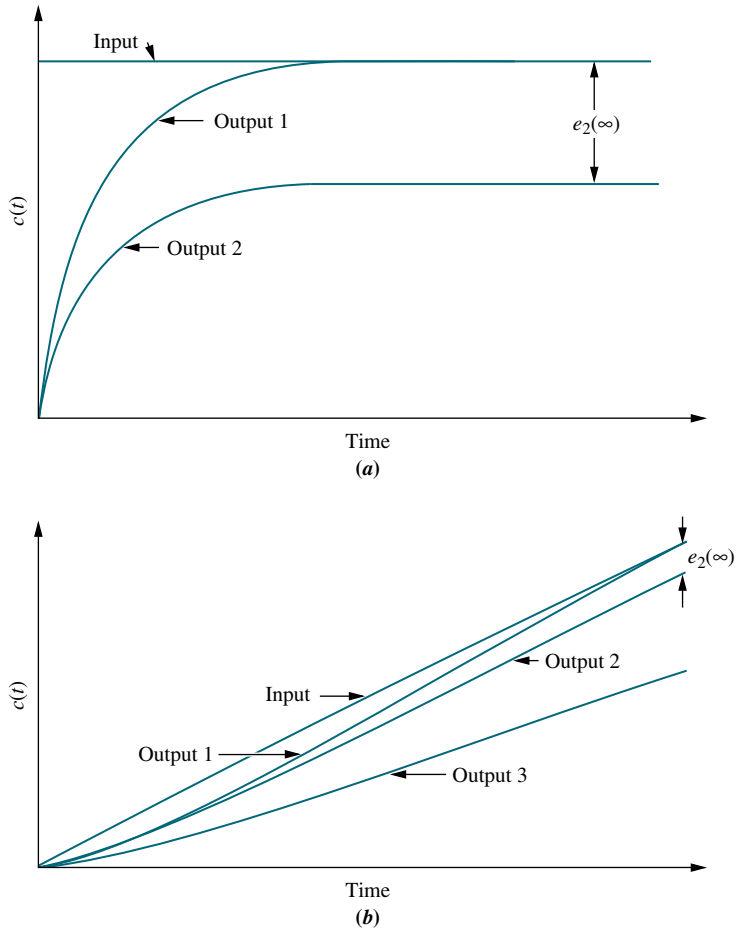


FIGURE 7.2 Steady-state error: **a.** step input; **b.** ramp input

another possibility exists. If the output's slope is different from that of the input, then output 3, shown in Figure 7.2(b), results. Here the steady-state error is infinite as measured vertically between the input and output 3 after the transients have died down, and t approaches infinity.

Let us now look at the error from the perspective of the most general block diagram. Since the error is the difference between the input and the output of a system, we assume a closed-loop transfer function, $T(s)$, and form the error, $E(s)$, by taking the difference between the input and the output, as shown in Figure 7.3(a). Here we are interested in the steady-state, or final, value of $e(t)$. For unity feedback systems, $E(s)$ appears as shown in Figure 7.3(b). In this chapter, we study and derive expressions for the steady-state error for unity feedback systems first and then expand to nonunity feedback systems. Before we begin our study of steady-state errors for unity feedback systems, let us look at the sources of the errors with which we deal.

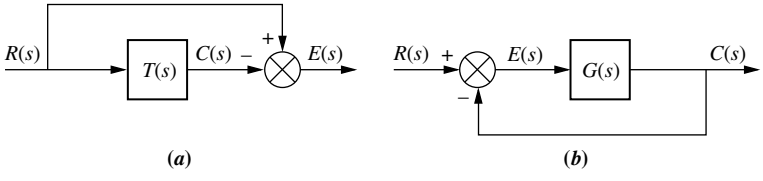


FIGURE 7.3 Closed-loop control system error: **a.** general representation; **b.** representation for unity feedback systems

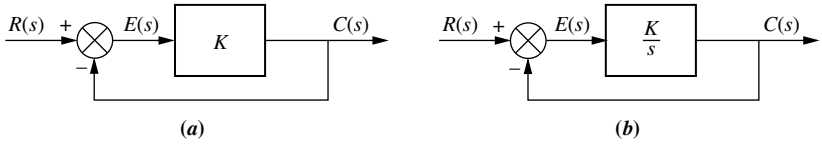


FIGURE 7.4 System with **a.** finite steady-state error for a step input; **b.** zero steady-state error for step input

Sources of Steady-State Error

Many steady-state errors in control systems arise from nonlinear sources, such as backlash in gears or a motor that will not move unless the input voltage exceeds a threshold. Nonlinear behavior as a source of steady-state errors, although a viable topic for study is beyond the scope of a text on linear control systems. The steady-state errors we study here are errors that arise from the configuration of the system itself and the type of applied input.

For example, look at the system of Figure 7.4(a), where $R(s)$ is the input, $C(s)$ is the output, and $E(s) = R(s) - C(s)$ is the error. Consider a step input. In the steady state, if $c(t)$ equals $r(t)$, $e(t)$ will be zero. But with a pure gain, K, the error, $e(t)$, cannot be zero if $c(t)$ is to be finite and nonzero. Thus, by virtue of the configuration of the system (a pure gain of K in the forward path), an error must exist. If we call $c_{\text{steady-state}}$ the steady-state value of the output and $e_{\text{steady-state}}$ the steady-state value of the error, then $c_{\text{steady-state}} = Ke_{\text{steady-state}}$, or

$$e_{\text{steady-state}} = \frac{1}{K} c_{\text{steady-state}} \tag{7.1}$$

Thus, the larger the value of K, the smaller the value of $e_{\text{steady-state}}$ would have to be to yield a similar value of $c_{\text{steady-state}}$. The conclusion we can draw is that with a pure gain in the forward path, there will always be a steady-state error for a step input. This error diminishes as the value of K increases.

If the forward-path gain is replaced by an integrator, as shown in Figure 7.4(b), there will be zero error in the steady state for a step input. The reasoning is as follows: As $c(t)$ increases, $e(t)$ will decrease, since $e(t) = r(t) - c(t)$. This decrease will continue until there is zero error, but there will still be a value for $c(t)$ since an integrator can have a constant output without any input. For example, a motor can be represented simply as an integrator. A voltage applied to the motor will cause rotation. When the applied voltage is removed, the motor will stop and remain at its present output position. Since it does not return to its initial position, we have an angular displacement output without an input to the motor. Therefore, a system similar to Figure 7.4(b), which uses a motor in the forward path, can have zero steady-state error for a step input.

We have examined two cases qualitatively to show how a system can be expected to exhibit various steady-state error characteristics, depending upon the system configuration. We now formalize the concepts and derive the relationships between the steady-state errors and the system configuration generating these errors.

7.2 Steady-State Error for Unity Feedback Systems

Steady-state error can be calculated from a system's closed-loop transfer function, $T(s)$, or the open-loop transfer function, $G(s)$, for unity feedback systems. We begin by deriving the system's steady-state error in terms of the closed-loop transfer

function, $T(s)$, in order to introduce the subject and the definitions. Next we obtain insight into the factors affecting steady-state error by using the open-loop transfer function, $G(s)$, in unity feedback systems for our calculations. Later in the chapter we generalize this discussion to nonunity feedback systems.

Steady-State Error in Terms of $T(s)$

Consider Figure 7.3(a). To find $E(s)$, the error between the input, $R(s)$, and the output, $C(s)$, we write

$$E(s) = R(s) - C(s) \tag{7.2}$$

But

$$C(s) = R(s)T(s) \tag{7.3}$$

Substituting Eq. (7.3) into Eq. (7.2), simplifying, and solving for $E(s)$ yields

$$E(s) = R(s)[1 - T(s)] \tag{7.4}$$

Although Eq. (7.4) allows us to solve for $e(t)$ at any time, t, we are interested in the final value of the error, $e(\infty)$. Applying the final value theorem,[1] which allows us to use the final value of $e(t)$ without taking the inverse Laplace transform of $E(s)$, and then letting t approach infinity, we obtain

$$e(\infty) = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) \tag{7.5}$$[2]

Substituting Eq. (7.4) into Eq. (7.5) yields

$$e(\infty) = \lim_{s\to\infty} sR(s)[1 - T(s)] \tag{7.6}$$

Let us look at an example.

---

[1] The final value theorem is derived from the Laplace transform of the derivative. Thus,

$$\mathscr{L}[\dot{f}(t)] = \int_{0-}^{\infty} \dot{f}(t)e^{st}dt = sF(s) - f(0-)$$

As $s \to 0$,

$$\int_{0-}^{\infty} \dot{f}(t)dt = f(\infty) - f(0-) = \lim_{s\to 0} sF(s) - f(0-)$$

or

$$f(\infty) = \lim_{s\to 0} sF(s)$$

For finite steady-state errors, the final value theorem is valid only if $F(s)$ has poles only in the left half-plane and, at most, one pole at the origin. However, correct results that yield steady-state errors that are infinite can be obtained if $F(s)$ has more than one pole at the origin (see *D'Azzo and Houpis, 1988*). If $F(s)$ has poles in the right half-plane or poles on the imaginary axis other than at the origin, the final value theorem is invalid.

[2] Valid only if (1) $E(s)$ has poles only in the left half-plane and at the origin, and (2) the closed-loop transfer function, $T(s)$, is stable. Notice that by using Eq. (7.5), numerical results can be obtained for unstable systems. These results, however, are meaningless.

Example 7.1

Steady-State Error in Terms of $T(s)$

PROBLEM: Find the steady-state error for the system of Figure 7.3(a) if $T(s) = 5/(s^2 + 7s + 10)$ and the input is a unit step.

SOLUTION: From the problem statement, $R(s) = 1/s$ and $T(s) = 5/(s^2 + 7s + 10)$. Substituting into Eq. (7.4) yields

$$E(s) = \frac{s^2 + 7s + 5}{s(s^2 + 7s + 10)} \tag{7.7}$$

Since $T(s)$ is stable and, subsequently, $E(s)$ does not have right–half-plane poles or $j\omega$ poles other than at the origin, we can apply the final value theorem. Substituting Eq. (7.7) into Eq. (7.5) gives $e(\infty) = 1/2$.

Steady-State Error in Terms of $G(s)$

Many times we have the system configured as a unity feedback system with a forward transfer function, $G(s)$. Although we can find the closed-loop transfer function, $T(s)$, and then proceed as in the previous subsection, we find more insight for analysis and design by expressing the steady-state error in terms of $G(s)$ rather than $T(s)$.

Consider the feedback control system shown in Figure 7.3(b). Since the feedback, $H(s)$, equals 1, the system has unity feedback. The implication is that $E(s)$ is actually the error between the input, $R(s)$, and the output, $C(s)$. Thus, if we solve for $E(s)$, we will have an expression for the error. We will then apply the final value theorem, Item 11 in Table 2.2, to evaluate the steady-state error.

Writing $E(s)$ from Figure 7.3(b), we obtain

$$E(s) = R(s) - C(s) \tag{7.8}$$

But

$$C(s) = E(s)G(s) \tag{7.9}$$

Finally, substituting Eq. (7.9) into Eq. (7.8) and solving for $E(s)$ yields

$$E(s) = \frac{R(s)}{1 + G(s)} \tag{7.10}$$

We now apply the final value theorem, Eq. (7.5). At this point in a numerical calculation, we must check to see whether the closed-loop system is stable, using, for example, the Routh-Hurwitz criterion. For now, though, assume that the closed-loop system is stable and substitute Eq. (7.10) into Eq. (7.5), obtaining

$$e(\infty) = \lim_{s \to 0} \frac{sR(s)}{1 + G(s)} \tag{7.11}$$

Equation (7.11) allows us to calculate the steady-state error, $e(\infty)$, given the input, $R(s)$, and the system, $G(s)$. We now substitute several inputs for $R(s)$ and then

draw conclusions about the relationships that exist between the open-loop system, $G(s)$, and the nature of the steady-state error, $e(\infty)$.

The three test signals we use to establish specifications for a control system's steady-state error characteristics are shown in Table 7.1. Let us take each input and evaluate its effect on the steady-state error by using Eq. (7.11).

Step Input. Using Eq. (7.11) with $R(s) = 1/s$, we find

$$e(\infty) = e_{\text{step}}(\infty) = \lim_{s\to 0} \frac{s(1/s)}{1 + G(s)} = \frac{1}{1 + \lim\limits_{s\to 0} G(s)} \tag{7.12}$$

The term

$$\lim_{s\to 0} G(s)$$

is the dc gain of the forward transfer function, since s, the frequency variable, is approaching zero. In order to have zero steady-state error,

$$\lim_{s\to 0} G(s) = \infty \tag{7.13}$$

Hence, to satisfy Eq. (7.13), $G(s)$ must take on the following form:

$$G(s) \equiv \frac{(s + z_1)(s + z_2)\cdots}{s^n(s + p_1)(s + p_2)\cdots} \tag{7.14}$$

and for the limit to be infinite, the denominator must be equal to zero as s goes to zero. Thus, $n \geq 1$; that is, at least one pole must be at the origin. Since division by s in the frequency domain is integration in the time domain (see Table 2.2, Item 10), we are also saying that at least one pure integration must be present in the forward path. The steady-state response for this case of zero steady-state error is similar to that shown in Figure 7.2(a), output 1.

If there are no integrations, then $n = 0$. Using Eq. (7.14), we have

$$\lim_{s\to 0} G(s) = \frac{z_1 z_2 \cdots}{p_1 p_2 \cdots} \tag{7.15}$$

which is finite and yields a finite error from Eq. (7.12). Figure 7.2(a), output 2, is an example of this case of finite steady-state error.

In summary, for a step input to a unity feedback system, the steady-state error will be zero if there is at least one pure integration in the forward path. If there are no integrations, then there will be a nonzero finite error. This result is comparable to our qualitative discussion in Section 7.1, where we found that a pure gain yields a constant steady-state error for a step input, but an integrator yields zero error for the same type of input. We now repeat the development for a ramp input.

Ramp Input. Using Eq. (7.11) with, $R(s) = 1/s^2$, we obtain

$$e(\infty) = e_{\text{ramp}}(\infty) = \lim_{s\to 0} \frac{s(1/s^2)}{1 + G(s)} = \lim_{s\to 0} \frac{1}{s + sG(s)} = \frac{1}{\lim\limits_{s\to 0} sG(s)} \tag{7.16}$$

To have zero steady-state error for a ramp input, we must have

$$\lim_{s\to 0} sG(s) = \infty \tag{7.17}$$

To satisfy Eq. (7.17), $G(s)$ must take the same form as Eq. (7.14), except that $n \geq 2$. In other words, there must be at least two integrations in the forward path. An example of zero steady-state error for a ramp input is shown in Figure 7.2(b), output 1.

If only one integration exists in the forward path, then, assuming Eq. (7.14),

$$\lim_{s\to 0} sG(s) = \frac{z_1 z_2 \cdots}{p_1 p_2 \cdots} \tag{7.18}$$

which is finite rather than infinite. Using Eq. (7.16), we find that this configuration leads to a constant error, as shown in Figure 7.2(b), output 2.

If there are no integrations in the forward path, then

$$\lim_{s\to 0} sG(s) = 0 \tag{7.19}$$

and the steady-state error would be infinite and lead to diverging ramps, as shown in Figure 7.2(b), output 3. Finally, we repeat the development for a parabolic input.

Parabolic Input. Using Eq. (7.11) with $R(s) = 1/s^3$, we obtain

$$e(\infty) = e_{\text{parabola}}(\infty) = \lim_{s\to 0} \frac{s(1/s^3)}{1 + G(s)} = \lim_{s\to 0} \frac{1}{s^2 + s^2 G(s)} = \frac{1}{\lim\limits_{s\to 0} s^2 G(s)} \tag{7.20}$$

In order to have zero steady-state error for a parabolic input, we must have

$$\lim_{s\to 0} s^2 G(s) = \infty \tag{7.21}$$

To satisfy Eq. (7.21), $G(s)$ must take on the same form as Eq. (7.14), except that $n \geq 3$. In other words, there must be at least three integrations in the forward path.

If there are only two integrations in the forward path, then

$$\lim_{s\to 0} s^2 G(s) = \frac{z_1 z_2 \cdots}{p_1 p_2 \cdots} \tag{7.22}$$

is finite rather than infinite. Using Eq. (7.20), we find that this configuration leads to a constant error.

If there is only one or less integration in the forward path, then

$$\lim_{s\to 0} s^2 G(s) = 0 \tag{7.23}$$

and the steady-state error is infinite. Two examples demonstrate these concepts.

Example 7.2

Steady-State Errors for Systems with No Integrations

PROBLEM: Find the steady-state errors for inputs of $5u(t)$, $5tu(t)$, and $5t^2u(t)$ to the system shown in Figure 7.5. The function $u(t)$ is the unit step.

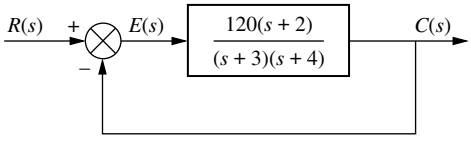


FIGURE 7.5 Feedback control system for Example 7.2

SOLUTION: First we verify that the closed-loop system is indeed stable. For this example we leave out the details. Next, for the input $5u(t)$, whose Laplace transform is $5/s$, the steady-state error will be five times as large as that given by Eq. (7.12), or

$$e(\infty) = e_{\text{step}}(\infty) = \frac{5}{1 + \lim_{s\to 0} G(s)} = \frac{5}{1+20} = \frac{5}{21} \tag{7.24}$$

which implies a response similar to output 2 of Figure 7.2(a).

For the input $5tu(t)$, whose Laplace transform is $5/s^2$, the steady-state error will be five times as large as that given by Eq. (7.16), or

$$e(\infty) = e_{\text{ramp}}(\infty) = \frac{5}{\lim_{s\to 0} sG(s)} = \frac{5}{0} = \infty \tag{7.25}$$

which implies a response similar to output 3 of Figure 7.2(b).

For the input $5t^2u(t)$, whose Laplace transform is $10/s^3$, the steady-state error will be 10 times as large as that given by Eq. (7.20), or

$$e(\infty) = e_{\text{parabola}}(\infty) = \frac{10}{\lim_{s\to 0} s^2G(s)} = \frac{10}{0} = \infty \tag{7.26}$$

Example 7.3

Steady-State Errors for Systems with One Integration

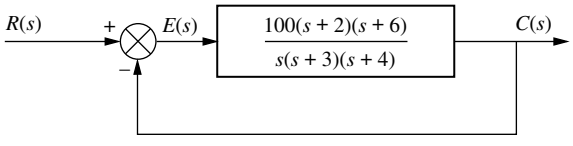


FIGURE 7.6 Feedback control system for Example 7.3

PROBLEM: Find the steady-state errors for inputs of $5u(t)$, $5tu(t)$, and $5t^2u(t)$ to the system shown in Figure 7.6. The function $u(t)$ is the unit step.

SOLUTION: First verify that the closed-loop system is indeed stable. For this example we leave out the details. Next note that since there is an integration in the forward path, the steady-state errors for some of the input waveforms will be less than those found in Example 7.2. For the input $5u(t)$, whose Laplace transform is $5/s$, the steady-state error will be five times as large as that given by Eq. (7.12), or

$$e(\infty) = e_{\text{step}}(\infty) = \frac{5}{1 + \lim_{s\to 0} G(s)} = \frac{5}{\infty} = 0 \tag{7.27}$$

which implies a response similar to output 1 of Figure 7.2(a). Notice that the integration in the forward path yields zero error for a step input, rather than the finite error found in Example 7.2.

For the input $5tu(t)$, whose Laplace transform is $5/s^2$, the steady-state error will be five times as large as that given by Eq. (7.16), or

$$e(\infty) = e_{\text{ramp}}(\infty) = \frac{5}{\lim_{s\to 0} sG(s)} = \frac{5}{100} = \frac{1}{20} \tag{7.28}$$

which implies a response similar to output 2 of Figure 7.2(*b*). Notice that the integration in the forward path yields a finite error for a ramp input, rather than the infinite error found in Example 7.2.

For the input, $5t^2u(t)$, whose Laplace transform is $10/s^3$, the steady-state error will be 10 times as large as that given by Eq. (7.20), or

$$e(\infty) = e_{\text{parabola}}(\infty) = \frac{10}{\lim_{s\to 0} s^2G(s)} = \frac{10}{0} = \infty \tag{7.29}$$

Notice that the integration in the forward path does not yield any improvement in steady-state error over that found in Example 7.2 for a parabolic input.

Skill-Assessment Exercise 7.1

PROBLEM: A unity feedback system has the following forward transfer function:

WileyPLUS
WPCS
Control Solutions

$$G(s) = \frac{10(s+20)(s+30)}{s(s+25)(s+35)}$$

a. Find the steady-state error for the following inputs: $15u(t)$, $15tu(t)$, and $15t^2u(t)$.

b. Repeat for

$$G(s) = \frac{10(s+20)(s+30)}{s^2(s+25)(s+35)(s+50)}$$

ANSWERS:

a. The closed-loop system is stable. For $15u(t)$, $e_{\text{step}}(\infty) = 0$; for $15tu(t)$, $e_{\text{ramp}}(\infty) = 2.1875$; for $15(t^2)u(t)$, $e_{\text{parabola}}(\infty) = \infty$.

b. The closed-loop system is unstable. Calculations cannot be made.

The complete solution is at www.wiley.com/college/nise.

7.3 Static Error Constants and System Type

We continue our focus on unity negative feedback systems and define parameters that we can use as steady-state error performance specifications, just as we defined damping ratio, natural frequency, settling time, percent overshoot, and so on as performance specifications for the transient response. These steady-state error performance specifications are called *static error constants*. Let us see how they are defined, how to calculate them, and, in the next section, how to use them for design.

Static Error Constants

In the previous section we derived the following relationships for steady-state error. For a step input, $u(t)$,

$$e(\infty) = e_{\text{step}}(\infty) = \frac{1}{1 + \lim_{s\to 0} G(s)} \tag{7.30}$$

For a ramp input, $tu(t)$,

$$e(\infty) = e_{\text{ramp}}(\infty) = \frac{1}{\lim_{s\to 0} sG(s)} \tag{7.31}$$

For a parabolic input, $\frac{1}{2}t^2u(t)$.

$$e(\infty) = e_{\text{parabola}}(\infty) = \frac{1}{\lim_{s\to 0} s^2G(s)} \tag{7.32}$$

The three terms in the denominator that are taken to the limit determine the steady-state error. We call these limits *static error constants.* Individually, their names are

position constant, K_p, where

$$K_p = \lim_{s\to 0} G(s) \tag{7.33}$$

velocity constant, K_v, where

$$K_v = \lim_{s\to 0} sG(s) \tag{7.34}$$

acceleration constant, K_a, where

$$K_a = \lim_{s\to 0} s^2G(s) \tag{7.35}$$

As we have seen, these quantities, depending upon the form of $G(s)$, can assume values of zero, finite constant, or infinity. Since the static error constant appears in the denominator of the steady-state error. Eqs. (7.30) through (7.32), the value of the steady-state error decreases as the static error constant increases.

In Section 7.2, we evaluated the steady-state error by using the final value theorem. An alternate method makes use of the static error constants. A few examples follow.

Example 7.4

Steady-State Error via Static Error Constants

PROBLEM: For each system of Figure 7.7, evaluate the static error constants and find the expected error for the standard step, ramp, and parabolic inputs.

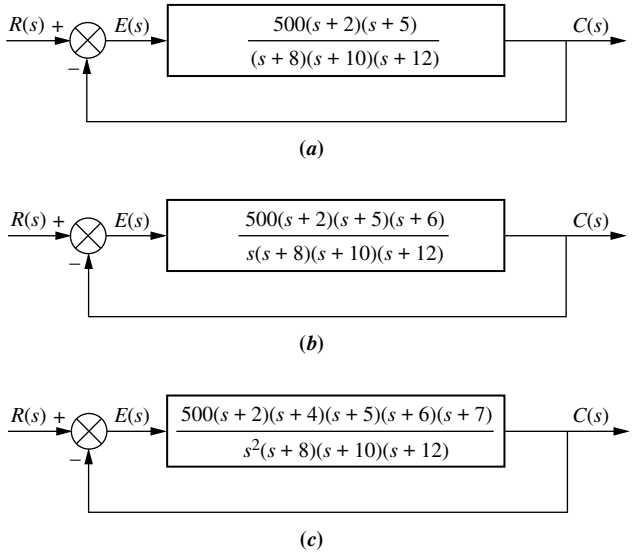


FIGURE 7.7 Feedback control systems for Example 7.4

SOLUTION: First verify that all closed-loop systems shown are indeed stable. For this example we leave out the details. Next, for Figure 7.7(a),

$$K_p = \lim_{s\to 0} G(s) = \frac{500 \times 2 \times 5}{8 \times 10 \times 12} = 5.208 \tag{7.36}$$

$$K_v = \lim_{s\to 0} sG(s) = 0 \tag{7.37}$$

$$K_a = \lim_{s\to 0} s^2 G(s) = 0 \tag{7.38}$$

Thus, for a step input,

$$e(\infty) = \frac{1}{1 + K_p} = 0.161 \tag{7.39}$$

For a ramp input,

$$e(\infty) = \frac{1}{K_v} = \infty \tag{7.40}$$

For a parabolic input,

$$e(\infty) = \frac{1}{K_a} = \infty \tag{7.41}$$

Now, for Figure 7.7(b),

$$K_p = \lim_{s\to 0} G(s) = \infty \tag{7.42}$$

$$K_v = \lim_{s\to 0} sG(s) = \frac{500 \times 2 \times 5 \times 6}{8 \times 10 \times 12} = 31.25 \tag{7.43}$$

and

$$K_a = \lim_{s\to 0} s^2 G(s) = 0 \tag{7.44}$$

Thus, for a step input,

$$e(\infty) = \frac{1}{1+K_p} = 0 \tag{7.45}$$

For a ramp input,

$$e(\infty) = \frac{1}{K_v} = \frac{1}{31.25} = 0.032 \tag{7.46}$$

For a parabolic input,

$$e(\infty) = \frac{1}{K_a} = \infty \tag{7.47}$$

Finally, for Figure 7.7(c),

$$K_p = \lim_{s\to 0} G(s) = \infty \tag{7.48}$$

$$K_v = \lim_{s\to 0} sG(s) = \infty \tag{7.49}$$

and

$$K_a = \lim_{s\to 0} s^2 G(s) = \frac{500\times 2\times 4\times 5\times 6\times 7}{8\times 10\times 12} = 875 \tag{7.50}$$

Thus, for a step input,

$$e(\infty) = \frac{1}{1+K_p} = 0 \tag{7.51}$$

For a ramp input,

$$e(\infty) = \frac{1}{K_v} = 0 \tag{7.52}$$

For a parabolic input,

$$e(\infty) = \frac{1}{K_a} = \frac{1}{875} = 1.14\times 10^{-3} \tag{7.53}$$

MATLAB
ML

Students who are using MATLAB should now run ch7p1 in Appendix B. You will learn how to test the system for stability, evaluate static error constants, and calculate steady-state error using MATLAB. This exercise applies MATLAB to solve Example 7.4 with System (b).

System Type

Let us continue to focus on a unity negative feedback system. The values of the static error constants, again, depend upon the form of $G(s)$, especially the number of pure integrations in the forward path. Since steady-state errors are dependent upon the

number of integrations in the forward path, we give a name to this system attribute. Given the system in Figure 7.8, we define *system type* to be the value of n in the denominator or, equivalently, the number of pure integrations in the forward path. Therefore, a system with $n = 0$ is a Type 0 system. If $n = 1$ or $n = 2$, the corresponding system is a Type 1 or Type 2 system, respectively.

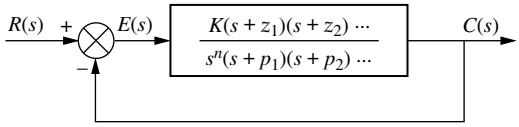


FIGURE 7.8 Feedback control system for defining system type

Table 7.2 ties together the concepts of steady-state error, static error constants, and system type. The table shows the static error constants and the steady-state errors as functions of input waveform and system type.

TABLE 7.2 Relationships between input, system type, static error constants, and steady-state errors

		Type 0		Type 1		Type 2	
Input	**Steady-state error formula**	**Static error constant**	**Error**	**Static error constant**	**Error**	**Static error constant**	**Error**
Step, $u(t)$	$\frac{1}{1+K_p}$	$K_p =$ Constant	$\frac{1}{1+K_p}$	$K_p = \infty$	0	$K_p = \infty$	0
Ramp, $tu(t)$	$\frac{1}{K_v}$	$K_v = 0$	∞	$K_v =$ Constant	$\frac{1}{K_v}$	$K_v = \infty$	0
Parabola, $\frac{1}{2}t^2u(t)$	$\frac{1}{K_a}$	$K_a = 0$	∞	$K_a = 0$	∞	$K_a =$ Constant	$\frac{1}{K_a}$

Skill-Assessment Exercise 7.2

PROBLEM: A unity feedback system has the following forward transfer function:

$$G(s) = \frac{1000(s+8)}{(s+7)(s+9)}$$

a. Evaluate system type, K_p, K_v, and K_a.

b. Use your answers to **a.** to find the steady-state errors for the standard step, ramp, and parabolic inputs.

ANSWERS:

a. The closed-loop system is stable. System type = Type 0. $K_p = 127$, $K_v = 0$, and $K_a = 0$.

b. $e_{\text{step}}(\infty) = 7.8 \times 10^{-3}$, $e_{\text{ramp}}(\infty) = \infty$, and $e_{\text{parabola}}(\infty) = \infty$

The complete solution is at www.wiley.com/college/nise.

TryIt 7.1

Use MATLAB, the Control System Toolbox, and the following statements to find K_p, $e_{\text{step}}(\infty)$, and the closed-loop poles to check for stability for the system of Skill-Assessment Exercise 7.2.

```
numg=1000*[1 8];
deng=poly([-7 -9]);
G=tf(numg,deng);
Kp=dcgain(G)
estep=1/(1+Kp)
T=feedback(G,1);
poles=pole(T)
```

In this section, we defined steady-state errors, static error constants, and system type. Now the specifications for a control system's steady-state errors will be formulated, followed by some examples.

7.4 Steady-State Error Specifications

Static error constants can be used to specify the steady-state error characteristics of control systems, such as that shown in Figure 7.9. Just as damping ratio, ζ, settling time, T_s, peak time, T_p, and percent overshoot, $\%OS$, are used as specifications for a



FIGURE 7.9 A robot used in the manufacturing of semiconductor random-access memories (RAMs) similar to those in personal computers. Steady-state error is an important design consideration for assembly-line robots.

control system's transient response, so the position constant, K_p, velocity constant, K_v, and acceleration constant, K_a, can be used as specifications for a control system's steady-state errors. We will soon see that a wealth of information is contained within the specification of a static error constant.

For example, if a control system has the specification $K_v = 1000$, we can draw several conclusions:

1. The system is stable.
2. The system is of Type 1, since only Type 1 systems have K_v's that are finite constants. Recall that $K_v = 0$ for Type 0 systems, whereas $K_v = \infty$ for Type 2 systems.
3. A ramp input is the test signal. Since K_v is specified as a finite constant, and the steady-state error for a ramp input is inversely proportional to K_v, we know the test input is a ramp.
4. The steady-state error between the input ramp and the output ramp is $1/K_v$ per unit of input slope.

Let us look at two examples that demonstrate analysis and design using static error constants.

Example 7.5

Interpreting the Steady-State Error Specification

PROBLEM: What information is contained in the specification $K_p = 1000$?

SOLUTION: The system is stable. The system is Type 0, since only a Type 0 system has a finite K_p. Type 1 and Type 2 systems have $K_p = \infty$. The input test signal is a step, since K_p is specified. Finally, the error per unit step is

$$e(\infty) = \frac{1}{1 + K_p} = \frac{1}{1 + 1000} = \frac{1}{1001} \tag{7.54}$$

Example 7.6

Gain Design to Meet a Steady-State Error Specification

PROBLEM: Given the control system in Figure 7.10, find the value of K so that there is 10% error in the steady state.

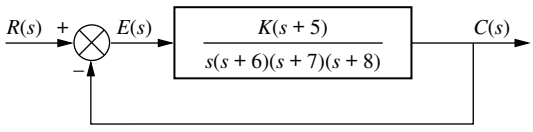


FIGURE 7.10 Feedback control system for Example 7.6

SOLUTION: Since the system is Type 1, the error stated in the problem must apply to a ramp input; only a ramp yields a finite error in a Type 1 system. Thus,

$$e(\infty) = \frac{1}{K_v} = 0.1 \tag{7.55}$$

Therefore,

$$K_v = 10 = \lim_{s\to 0} sG(s) = \frac{K \times 5}{6 \times 7 \times 8} \tag{7.56}$$

which yields

$$K = 672 \tag{7.57}$$

Applying the Routh-Hurwitz criterion, we see that the system is stable at this gain.

Although this gain meets the criteria for steady-state error and stability, it may not yield a desirable transient response. In Chapter 9 we will design feedback control systems to meet all three specifications.

MATLAB
ML

Students who are using MATLAB should now run ch7 p2 in Appendix B. You will learn how to find the gain to meet a steady-state error specification using MATLAB. This exercise solves Example 7.6 using MATLAB.

Skill-Assessment Exercise 7.3

PROBLEM: A unity feedback system has the following forward transfer function:

$$G(s) = \frac{K(s+12)}{(s+14)(s+18)}$$

Find the value of K to yield a 10% error in the steady state.

ANSWER: $K = 189$

The complete solution is at www.wiley.com/college/nise.

WileyPLUS
WPCS
Control Solutions

TryIt 7.2

Use MATLAB, the Control System Toolbox, and the following statements to solve Skill-Assessment Exercise 7.3 and check the resulting system for stability.

```
numg=[1 12];
deng=poly([-14 -18]);
G=tf(numg,deng);
Kpdk=dcgain(G);
estep=0.1;
K=(1/estep-1)/Kpdk
T=feedback(G,1);
poles=pole(T)
```

This example and exercise complete our discussion of unity feedback systems. In the remaining sections, we will deal with the steady-state errors for disturbances and the steady-state errors for feedback control systems in which the feedback is not unity.

7.5 Steady-State Error for Disturbances

Feedback control systems are used to compensate for disturbances or unwanted inputs that enter a system. The advantage of using feedback is that regardless of these disturbances, the system can be designed to follow the input with small or zero error, as we now demonstrate. Figure 7.11 shows a feedback control system with a disturbance, $D(s)$, injected between the controller and the plant. We now re-derive the expression for steady-state error with the disturbance included.

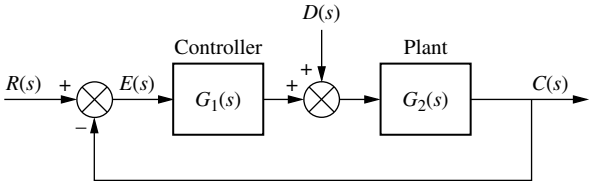


FIGURE 7.11 Feedback control system showing disturbance

The transform of the output is given by

$$C(s) = E(s)G_1(s)G_2(s) + D(s)G_2(s) \tag{7.58}$$

But

$$C(s) = R(s) - E(s) \tag{7.59}$$

Substituting Eq. (7.59) into Eq. (7.58) and solving for $E(s)$, we obtain

$$E(s) = \frac{1}{1 + G_1(s)G_2(s)}R(s) - \frac{G_2(s)}{1 + G_1(s)G_2(s)}D(s) \tag{7.60}$$

where we can think of $1/[1 + G_1(s)G_2(s)]$ as a transfer function relating $E(s)$ to $R(s)$ and $-G_2(s)/[1 + G_1(s)G_2(s)]$ as a transfer function relating $E(s)$ to $D(s)$.

To find the steady-state value of the error, we apply the final value theorem[3] to Eq. (7.60) and obtain

$$\begin{aligned} e(\infty) = \lim_{s\to 0} sE(s) &= \lim_{s\to 0} \frac{s}{1 + G_1(s)G_2(s)}R(s) - \lim_{s\to 0} \frac{sG_2(s)}{1 + G_1(s)G_2(s)}D(s) \\ &= e_R(\infty) + e_D(\infty) \end{aligned} \tag{7.61}$$

where

$$e_R(\infty) = \lim_{s\to 0} \frac{s}{1 + G_1(s)G_2(s)}R(s)$$

and

$$e_D(\infty) = -\lim_{s\to 0} \frac{sG_2(s)}{1 + G_1(s)G_2(s)}D(s)$$

The first term, $e_R(\infty)$, is the steady-state error due to $R(s)$, which we have already obtained. The second term, $e_D(\infty)$, is the steady-state error due to the disturbance. Let us explore the conditions on $e_D(\infty)$ that must exist to reduce the error due to the disturbance.

At this point, we must make some assumptions about $D(s)$, the controller, and the plant. First we assume a step disturbance, $D(s) = 1/s$. Substituting this value into

[3] Remember that the final value theorem can be applied only if the system is stable, with the roots of $[1 + G_1(s)G_2(s)]$ in the left–half–plane.

the second term of Eq. (7.61), $e_D(\infty)$, the steady-state error component due to a step disturbance is found to be

$$e_D(\infty) = -\frac{1}{\lim_{s\to 0}\frac{1}{G_2(s)} + \lim_{s\to 0} G_1(s)} \tag{7.62}$$

This equation shows that the steady-state error produced by a step disturbance can be reduced by increasing the dc gain of $G_1(s)$ or decreasing the dc gain of $G_2(s)$.

This concept is shown in Figure 7.12, where the system of Figure 7.11 has been rearranged so that the disturbance, $D(s)$, is depicted as the input and the error, $E(s)$, as the output, with $R(s)$ set equal to zero. If we want to minimize the steady-state value of $E(s)$, shown as the output in Figure 7.12, we must either increase the dc gain of $G_1(s)$ so that a lower value of $E(s)$ will be fed back to match the steady-state value of $D(s)$, or decrease the dc value of $G_2(s)$, which then yields a smaller value of $e(\infty)$ as predicted by the feedback formula.

Let us look at an example and calculate the numerical value of the steady-state error that results from a disturbance.

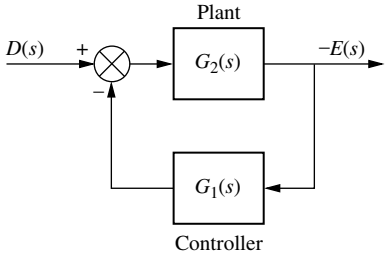


FIGURE 7.12 Figure 7.11 system rearranged to show disturbance as input and error as output, with $R(s) = 0$

Example 7.7

Steady-State Error Due to Step Disturbance

PROBLEM: Find the steady-state error component due to a step disturbance for the system of Figure 7.13.

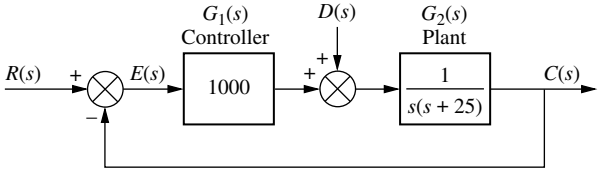


FIGURE 7.13 Feedback control system for Example 7.7

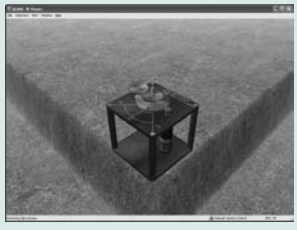
SOLUTION: The system is stable. Using Figure 7.12 and Eq. (7.62), we find

$$e_D(\infty) = -\frac{1}{\lim_{s\to 0}\frac{1}{G_2(s)} + \lim_{s\to 0} G_1(s)} = -\frac{1}{0 + 1000} = -\frac{1}{1000} \tag{7.63}$$

The result shows that the steady-state error produced by the step disturbance is inversely proportional to the dc gain of $G_1(s)$. The dc gain of $G_2(s)$ is infinite in this example.

Virtual Experiment 7.1
Steady-State Error

Put theory into practice finding the steady-state error of the Quanser Rotary Servo when subject to an input or a disturbance by simulating it in LabVIEW. This analysis becomes important when developing controllers for bottle labelling machines or robot joint control.



Virtual experiments are found on WileyPLUS.

Skill-Assessment Exercise 7.4

PROBLEM: Evaluate the steady-state error component due to a step disturbance for the system of Figure 7.14.

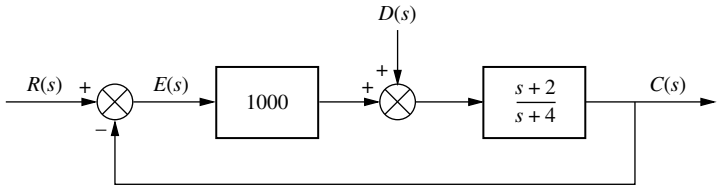


FIGURE 7.14 System for Skill-Assessment Exercise 7.4

ANSWER: $e_D(\infty) = -9.98 \times 10^{-4}$

The complete solution is at www.wiley.com/college/nise.

7.6 Steady-State Error for Nonunity Feedback Systems

Control systems often do not have unity feedback because of the compensation used to improve performance or because of the physical model for the system. The feedback path can be a pure gain other than unity or have some dynamic representation.

A general feedback system, showing the input transducer, $G_1(s)$, controller and plant, $G_2(s)$, and feedback, $H_1(s)$, is shown in Figure 7.15(a). Pushing the input transducer

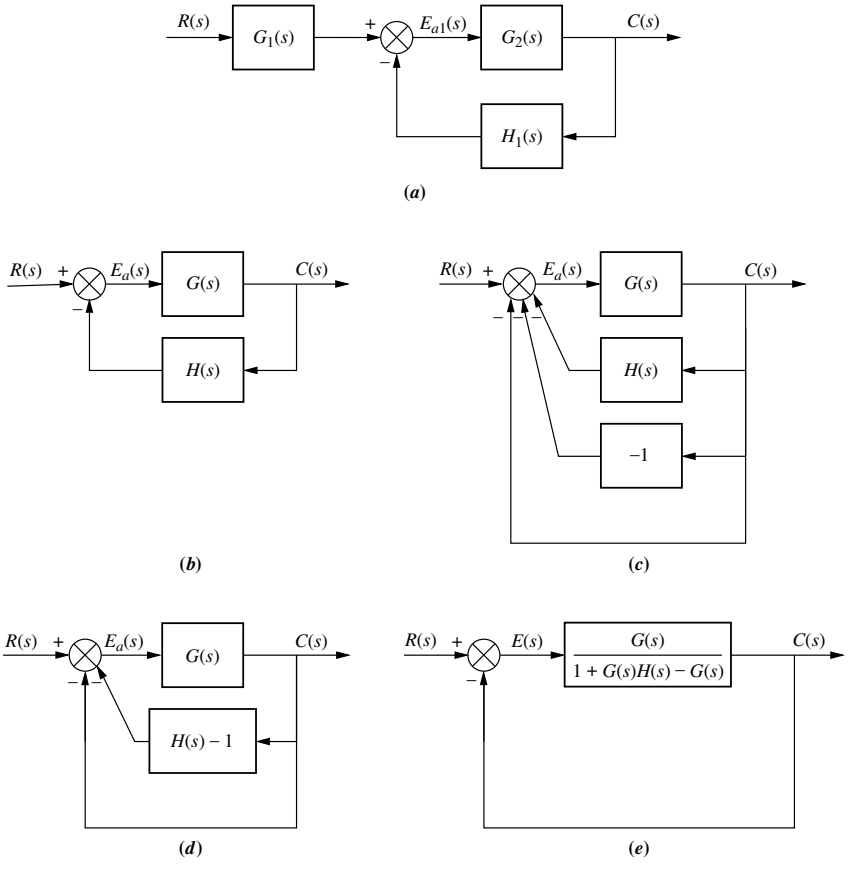


FIGURE 7.15 Forming an equivalent unity feedback system from a general nonunity feedback system

to the right past the summing junction yields the general nonunity feedback system shown in Figure 7.15(b), where $G(s) = G_1(s)G_2(s)$ and $H(s) = H_1(s)/G_1(s)$. Notice that unlike a unity feedback system, where $H(s) = 1$, the error is not the difference between the input and the output. For this case we call the signal at the output of the summing junction the *actuating signal,* $E_a(s)$. If $r(t)$ and $c(t)$ have the same units, we can find the steady-state error, $e(\infty) = r(\infty) - c(\infty)$. The first step is to show explicitly $E(s) = R(s) - C(s)$ on the block diagram.

Take the nonunity feedback control system shown in Figure 7.15(b) and form a unity feedback system by adding and subtracting unity feedback paths, as shown in Figure 7.15(c). This step requires that input and output units be the same. Next combine $H(s)$ with the negative unity feedback, as shown in Figure 7.15(d). Finally, combine the feedback system consisting of $G(s)$ and $[H(s) - 1]$, leaving an equivalent forward path and a unity feedback, as shown in Figure 7.15(e). Notice that the final figure shows $E(s) = R(s) - C(s)$ explicitly.

The following example summarizes the concepts of steady-state error, system type, and static error constants for nonunity feedback systems.

Example 7.8

Steady-State Error for Nonunity Feedback Systems

PROBLEM: For the system shown in Figure 7.16, find the system type, the appropriate error constant associated with the system type, and the steady-state error for a unit step input. Assume input and output units are the same.

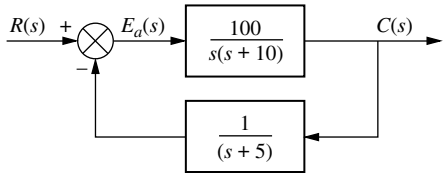


FIGURE 7.16 Nonunity feedback control system for Example 7.8

SOLUTION: After determining that the system is indeed stable, one may impulsively declare the system to be Type 1. This may not be the case, since there is a nonunity feedback element, and the plant's actuating signal is not the difference between the input and the output. The first step in solving the problem is to convert the system of Figure 7.16 into an equivalent unity feedback system. Using the equivalent forward transfer function of Figure 7.15(e) along with

$$G(s) = \frac{100}{s(s+10)} \tag{7.64}$$

and

$$H(s) = \frac{1}{(s+5)} \tag{7.65}$$

we find

$$G_e(s) = \frac{G(s)}{1 + G(s)H(s) - G(s)} = \frac{100(s+5)}{s^3 + 15s^2 - 50s - 400} \tag{7.66}$$

Thus, the system is Type 0, since there are no pure integrations in Eq. (7.66). The appropriate static error constant is then K_p, whose value is

$$K_p = \lim_{s \to 0} G_e(s) = \frac{100 \times 5}{-400} = -\frac{5}{4} \tag{7.67}$$

TryIt 7.3

Use MATLAB, the Control System Toolbox, and the following statements to find $G_e(s)$ in Example 7.8.

```
G=zpk([],[0 -10],100);
H=zpk([],-5,1);
Ge=feedback...
 (G,(H-1));
'Ge(s)'
Ge=tf(Ge)
T=feedback(Ge,1);
'Poles of T(s)'
pole(T)
```

The steady-state error, $e(\infty)$, is

$$e(\infty) = \frac{1}{1 + K_p} = \frac{1}{1 - (5/4)} = -4 \tag{7.68}$$

The negative value for steady-state error implies that the output step is larger than the input step.

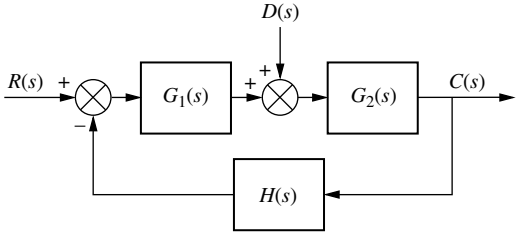


FIGURE 7.17 Nonunity feedback control system with disturbance

To continue our discussion of steady-state error for systems with nonunity feedback, let us look at the general system of Figure 7.17, which has both a disturbance and nonunity feedback. We will derive a general equation for the steady-state error and then determine the parameters of the system in order to drive the error to zero for step inputs and step disturbances.[4]

The steady-state error for this system, $e(\infty) = r(\infty) - c(\infty)$, is

$$\begin{aligned} e(\infty) = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\Bigg\{ &\left[1 - \frac{G_1(s)G_2(s)}{1 + G_1(s)G_2(s)H(s)}\right] R(s) \\ &- \left[\frac{G_2(s)}{1 + G_1(s)G_2(s)H(s)} D(s)\right]\Bigg\} \end{aligned} \tag{7.69}$$

Now limiting the discussion to step inputs and step disturbances, where $R(s) = D(s) = 1/s$, Eq. (7.69) becomes

$$\begin{aligned} e(\infty) = \lim_{s\to 0} sE(s) = \Bigg\{ &\left[1 - \frac{\lim_{s\to 0}[G_1(s)G_2(s)]}{\lim_{s\to 0}[1 + G_1(s)G_2(s)H(s)]}\right] \\ &- \left[\frac{\lim_{s\to 0} G_2(s)}{\lim_{s\to 0}[1 + G_1(s)G_2(s)H(s)]}\right]\Bigg\} \end{aligned} \tag{7.70}$$

For zero error,

$$\frac{\lim_{s\to 0}[G_1(s)G_2(s)]}{\lim_{s\to 0}[1 + G_1(s)G_2(s)H(s)]} = 1 \quad \text{and} \quad \frac{\lim_{s\to 0} G_2(s)}{\lim_{s\to 0}[1 + G_1(s)G_2(s)H(s)]} = 0 \tag{7.71}$$

The two equations in Eq. (7.71) can always be satisfied if (1) the system is stable, (2) $G_1(s)$ is a Type 1 system, (3) $G_2(s)$ is a Type 0 system, and (4) $H(s)$ is a Type 0 system with a dc gain of unity.

To conclude this section, we discuss finding the steady-state value of the actuating signal, $E_{a1}(s)$, in Figure 7.15(a). For this task there is no restriction that the input and output units be the same, since we are finding the steady-state difference between signals at the summing junction, which do have the same units.[5] The steady-state

[4] Details of the derivation are included as a problem at the end of this chapter.

[5] For clarity, steady-state error is the steady-state difference between the input and the output. Steady-state actuating signal is the steady-state difference at the output of the summing junction. In questions asking for steady-state error in problems, examples, and skill-assessment exercises, it will be assumed that input and output units are the same.

actuating signal for Figure 7.15(*a*) is

$$e_{a1}(\infty) = \lim_{s\to 0} \frac{sR(s)G_1(s)}{1 + G_2(s)H_1(s)} \tag{7.72}$$

The derivation is left to the student in the problem set at the end of this chapter.

Example 7.9

Steady-State Actuating Signal for Nonunity Feedback Systems

PROBLEM: Find the steady-state actuating signal for the system of Figure 7.16 for a unit step input. Repeat for a unit ramp input.

SOLUTION: Use Eq. (7.72) with $R(s) = 1/s$, a unit step input, $G_1(s) = 1$, $G_2(s) = 100/[s(s + 10)]$, and $H_1(s) = 1/(s + 5)$. Also, realize that $e_{a1}(\infty) = e_a(\infty)$, since $G_1(s) = 1$. Thus,

$$e_a(\infty) = \lim_{s\to 0} \frac{s\left(\frac{1}{s}\right)}{1 + \left(\frac{100}{s(s+10)}\right)\left(\frac{1}{(s+5)}\right)} = 0 \tag{7.73}$$

Now use Eq. (7.72) with $R(s) = 1/s^2$, a unit ramp input, and obtain

$$e_a(\infty) = \lim_{s\to 0} \frac{s\left(\frac{1}{s^2}\right)}{1 + \left(\frac{100}{s(s+10)}\right)\left(\frac{1}{(s+5)}\right)} = \frac{1}{2} \tag{7.74}$$

Skill-Assessment Exercise 7.5

PROBLEM:

a. Find the steady-state error, $e(\infty) = r(\infty) - c(\infty)$, for a unit step input given the nonunity feedback system of Figure 7.18. Repeat for a unit ramp input. Assume input and output units are the same.

b. Find the steady-state actuating signal, $e_a(\infty)$, for a unit step input given the nonunity feedback system of Figure 7.18. Repeat for a unit ramp input.

WileyPLUS
WPCS
Control Solutions

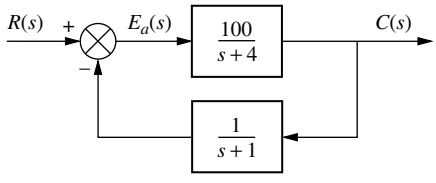


FIGURE 7.18 Nonunity feedback system for Skill-Assessment Exercise 7.5

ANSWERS:

a. $e_{\text{step}}(\infty) = 3.846 \times 10^{-2}$; $e_{\text{ramp}}(\infty) = \infty$

b. For a unit step input, $e_a(\infty) = 3.846 \times 10^{-2}$; for a unit ramp input, $e_a(\infty) = \infty$

The complete solution is at www.wiley.com/college/nise.

In this section, we have applied steady-state error analysis to nonunity feedback systems. When nonunity feedback is present, the plant's actuating signal is not

the actual error or difference between the input and the output. With nonunity feedback we may choose to (1) find the steady-state error for systems where the input and output units are the same or (2) find the steady-state actuating signal.

We also derived a general expression for the steady-state error of a nonunity feedback system with a disturbance. We used this equation to determine the attributes of the subsystems so that there was zero error for step inputs and step disturbances.

Before concluding this chapter, we will discuss a topic that is not only significant for steady-state errors but generally useful throughout the control systems design process.

7.7 Sensitivity

During the design process, the engineer may want to consider the extent to which changes in system parameters affect the behavior of a system. Ideally, parameter changes due to heat or other causes should not appreciably affect a system's performance. The degree to which changes in system parameters affect system transfer functions, and hence performance, is called *sensitivity*. A system with zero sensitivity (that is, changes in the system parameters have no effect on the transfer function) is ideal. The greater the sensitivity, the less desirable the effect of a parameter change.

For example, assume the function $F = K/(K + a)$. If $K = 10$ and $a = 100$, then $F = 0.091$. If parameter a triples to 300, then $F = 0.032$. We see that a fractional change in parameter a of $(300 - 100)/100 = 2$ (a 200% change), yields a change in the function F of $(0.032 - 0.091)/0.091 = -0.65$ (-65% change). Thus, the function F has reduced sensitivity to changes in parameter a. As we proceed, we will see that another advantage of feedback is that in general it affords reduced sensitivity to parameter changes.

Based upon the previous discussion, let us formalize a definition of sensitivity: *Sensitivity* is the ratio of the fractional change in the function to the fractional change in the parameter as the fractional change of the parameter approaches zero. That is,

$$\begin{aligned} S_{F:P} &= \lim_{\Delta P \to 0} \frac{\text{Fractional change in the function, } F}{\text{Fractional change in the parameter, } P} \\ &= \lim_{\Delta P \to 0} \frac{\Delta F/F}{\Delta P/P} \\ &= \lim_{\Delta P \to 0} \frac{P \Delta F}{F \Delta P} \end{aligned}$$

which reduces to

$$S_{F:P} = \frac{P}{F}\frac{\delta F}{\delta P} \tag{7.75}$$

Let us now apply the definition, first to a closed-loop transfer function and then to the steady-state error.

Example 7.10

Sensitivity of a Closed-Loop Transfer Function

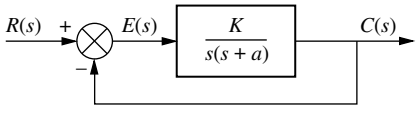


FIGURE 7.19 Feedback control system for Examples 7.10 and 7.11

PROBLEM: Given the system of Figure 7.19, calculate the sensitivity of the closed-loop transfer function to changes in the parameter a. How would you reduce the sensitivity?

SOLUTION: The closed-loop transfer function is

$$T(s) = \frac{K}{s^2 + as + K} \tag{7.76}$$

Using Eq. (7.75), the sensitivity is given by

$$S_{T:a} = \frac{a}{T}\frac{\delta T}{\delta a} = \frac{a}{\left(\dfrac{K}{s^2 + as + K}\right)}\left(\frac{-Ks}{(s^2 + as + K)^2}\right) = \frac{-as}{s^2 + as + K} \tag{7.77}$$

which is, in part, a function of the value of s. For any value of s, however, an increase in K reduces the sensitivity of the closed-loop transfer function to changes in the parameter a.

Example 7.11

Sensitivity of Steady-State Error with Ramp Input

PROBLEM: For the system of Figure 7.19, find the sensitivity of the steady-state error to changes in parameter K and parameter a with ramp inputs.

SOLUTION: The steady-state error for the system is

$$e(\infty) = \frac{1}{K_v} = \frac{a}{K} \tag{7.78}$$

The sensitivity of $e(\infty)$ to changes in parameter a is

$$S_{e:a} = \frac{a}{e}\frac{\delta e}{\delta a} = \frac{a}{a/K}\left[\frac{1}{K}\right] = 1 \tag{7.79}$$

The sensitivity of $e(\infty)$ to changes in parameter K is

$$S_{e:K} = \frac{K}{e}\frac{\delta e}{\delta K} = \frac{K}{a/K}\left[\frac{-a}{K^2}\right] = -1 \tag{7.80}$$

Thus, changes in either parameter a or parameter K are directly reflected in $e(\infty)$, and there is no reduction or increase in sensitivity. The negative sign in Eq. (7.80) indicates a decrease in $e(\infty)$ for an increase in K. Both of these results could have been obtained directly from Eq. (7.78) since $e(\infty)$ is directly proportional to parameter a and inversely proportional to parameter K.

Example 7.12

Sensitivity of Steady-State Error with Step Input

PROBLEM: Find the sensitivity of the steady-state error to changes in parameter K and parameter a for the system shown in Figure 7.20 with a step input.

SOLUTION: The steady-state error for this Type 0 system is

$$e(\infty) = \frac{1}{1 + K_p} = \frac{1}{1 + \dfrac{K}{ab}} = \frac{ab}{ab + K} \tag{7.81}$$

The sensitivity of $e(\infty)$ to changes in parameter a is

$$S_{e:a} = \frac{a}{e}\frac{\delta e}{\delta a} = \frac{a}{\left(\dfrac{ab}{ab + K}\right)} \quad \frac{(ab + K)b - ab^2}{(ab + K)^2} = \frac{K}{ab + K} \tag{7.82}$$

The sensitivity of $e(\infty)$ to changes in parameter K is

$$S_{e:K} = \frac{K}{e}\frac{\delta e}{\delta K} = \frac{K}{\left(\dfrac{ab}{ab + K}\right)} \quad \frac{-ab}{(ab + K)^2} = \frac{-K}{ab + K} \tag{7.83}$$

Equations (7.82) and (7.83) show that the sensitivity to changes in parameter K and parameter a is less than unity for positive a and b. Thus, feedback in this case yields reduced sensitivity to variations in both parameters.

Skill-Assessment Exercise 7.6

$$\frac{K(s + 7)}{s^2 + 2s + 10}$$

PROBLEM: Find the sensitivity of the steady-state error to changes in K for the system of Figure 7.21.

ANSWER: $S_{e:k} = \dfrac{-7K}{10 + 7K}$

The complete solution is at www.wiley.com/college/nise.

In this section, we defined sensitivity and showed that in some cases feedback reduces the sensitivity of a system's steady-state error to changes in system parameters. The concept of sensitivity can be applied to other measures of control system performance, as well; it is not limited to the sensitivity of the steady-state error performance.

7.8 Steady-State Error for Systems in State Space

Up to this point, we have evaluated the steady-state error for systems modeled as transfer functions. In this section, we will discuss how to evaluate the steady-state error for systems represented in state space. Two methods for calculating the steady-state error will be covered: (1) analysis via final value theorem and (2) analysis via input substitution. We will consider these methods individually.

Analysis via Final Value Theorem

A single-input, single-output system represented in state space can be analyzed for steady-state error using the final value theorem and the closed-loop transfer

supervisory functions consist of scheduling tasks, monitoring parameters and variables for out-of-range values, or initiating safety shutdown. Control functions are of primary interest to us, since a computer that performs within the feedback loop replaces the methods of compensation heretofore discussed. Examples of control functions are lead and lag compensation.

Transfer functions, representing compensators built with analog components, are now replaced with a digital computer that performs calculations that emulate the physical compensator. What advantages are there to replacing analog components with a digital computer?

Advantages of Digital Computers

The use of digital computers in the loop yields the following advantages over analog systems: (1) reduced cost, (2) flexibility in response to design changes, and (3) noise immunity. Modern control systems require control of numerous loops at the same time—pressure, position, velocity, and tension, for example. In the steel industry, a single digital computer can replace numerous analog controllers with a subsequent reduction in cost. Where analog controllers implied numerous adjustments and resulting hardware, digital systems are now installed. Banks of equipment, meters, and knobs are replaced with computer terminals, where information about settings and performance is obtained through menus and screen displays. Digital computers in the loop can yield a degree of flexibility in response to changes in design. Any changes or modifications that are required in the future can be implemented with simple software changes rather than expensive hardware modifications. Finally, digital systems exhibit more noise immunity than analog systems by virtue of the methods of implementation.

Where then is the computer placed in the loop? Remember that the digital computer is controlling numerous loops; thus, its position in the loop depends upon the function it performs. Typically, the computer replaces the cascade compensator and is thus positioned at the place shown in Figure 13.2(a).

The signals r, e, f, and c shown in Figure 13.2(a) can take on two forms: digital or analog. Up to this point we have used analog signals exclusively. Digital signals, which consist of a sequence of binary numbers, can be found in loops containing digital computers.

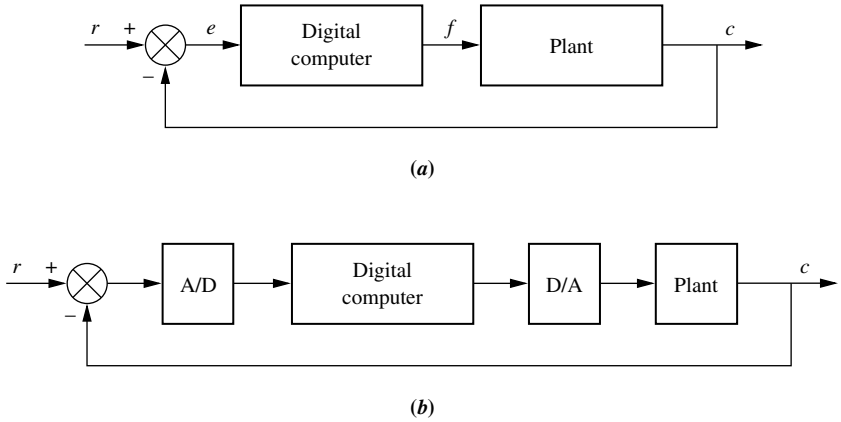


FIGURE 13.2 **a.** Placement of the digital computer within the loop; **b.** detailed block diagram showing placement of A/D and D/A converters

Loops containing both analog and digital signals must provide a means for conversion from one form to the other as required by each subsystem. A device that converts analog signals to digital signals is called an *analog-to-digital* (*A/D*) *converter*. Conversely, a device that converts digital signals to analog signals is called a *digital-to-analog* (*D/A*) *converter*. For example, in Figure 13.2(*b*), if the plant output, *c*, and the system input, *r*, are analog signals, then an analog-to-digital converter must be provided at the input to the digital computer. Also, if the plant input, *f*, is an analog signal, then a digital-to-analog converter must be provided at the output of the digital computer.

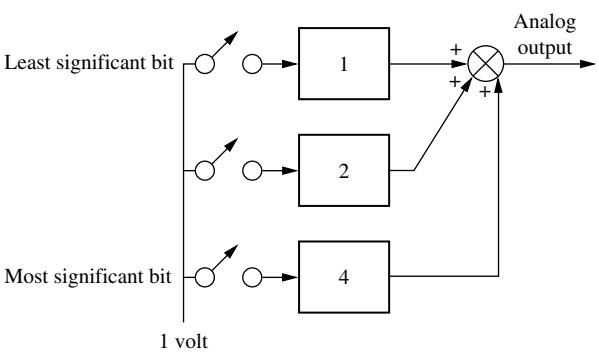


FIGURE 13.3 Digital-to-analog converter

Digital-to-Analog Conversion

Digital-to-analog conversion is simple and effectively instantaneous. Properly weighted voltages are summed together to yield the analog output. For example, in Figure 13.3, three weighted voltages are summed. The three-bit binary code is represented by the switches. Thus, if the binary number is 110_2, the center and bottom switches are on, and the analog output is 6 volts. In actual use, the switches are electronic and are set by the input binary code.

Analog-to-Digital Conversion

Analog-to-digital conversion, on the other hand, is a two-step process and is not instantaneous. There is a delay between the input analog voltage and the output digital word. In an analog-to-digital converter, the analog signal is first converted to a sampled signal and then converted to a sequence of binary numbers, the digital signal.

The sampling rate must be at least twice the bandwidth of the signal, or else there will be distortion. This minimum sampling frequency is called the *Nyquist sampling rate.*[1]

In Figure 13.4(*a*), we start with the analog signal. In Figure 13.4(*b*), we see the analog signal sampled at periodic intervals and held over the sampling interval by a device called a *zero-order sample-and-hold* (*z.o.h.*) that yields a staircase approximation to the analog signal. Higher-order holds, such as a first-order hold, generate more complex and more accurate waveshapes between samples. For example, a first-order hold generates a ramp between the samples. Samples are held before being digitized because the analog-to-digital converter converts the voltage to a digital number via a digital counter, which takes time to reach the correct digital number. Hence, the constant analog voltage must be present during the conversion process.

After sampling and holding, the analog-to-digital converter converts the sample to a digital number (as shown in Figure 13.4(*c*)), which is arrived at in the following manner. The dynamic range of the analog signal's voltage is divided into discrete levels, and each level is assigned a digital number. For example, in Figure 13.4(*b*), the analog signal is divided into eight levels. A three-bit digital number can represent each of the eight levels as shown in the figure. Thus, the difference between quantization levels is $M/8$ volts, where M is the maximum analog voltage. In general, for any system, this difference is $M/2^n$ volts, where n is the number of binary bits used for the analog-to-digital conversion.

Looking at Figure 13.4(*b*), we can see that there will be an associated error for each digitized analog value except the voltages at the boundaries such as $M/8$ and $2M/8$. We call this error the *quantization error.* Assuming that the quantization process rounds off the analog voltage to the next higher or lower level, the maximum

[1] See *Ogata (1987: 170–177)* for a detailed discussion.

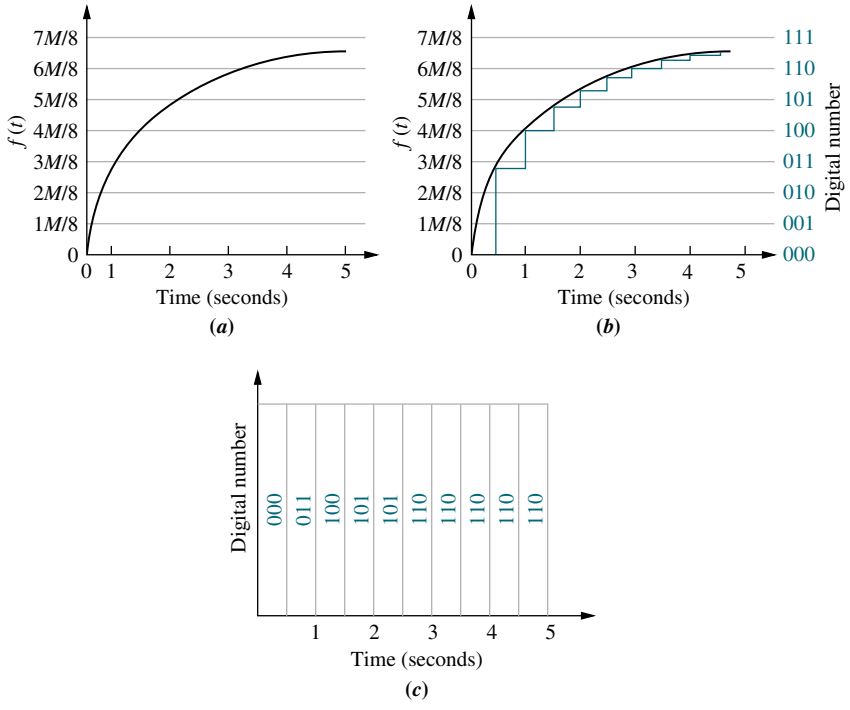


FIGURE 13.4 Steps in analog-to-digital conversion: **a.** analog signal; **b.** analog signal after sample-and-hold; **c.** conversion of samples to digital numbers

value of the quantization error is $1/2$ the difference between quantization levels in the range of analog voltages from 0 to $15M/16$. In general, for any system using roundoff, the quantization error will be $(1/2)(M/2^n) = M/2^{n+1}$.

We have now covered the basic concepts of digital systems. We found out why they are used, where the digital computer is placed in the loop, and how to convert between analog and digital signals. Since the computer can replace the compensator, we have to realize that the computer is working with a quantized amplitude representation of the analog signal formed from values of the analog signal at discrete intervals of time. Ignoring the quantization error, we see that the computer performs just as the compensator does, except that signals pass through the computer only at the sampled intervals of time. We will find that the sampling of data has an unusual effect upon the performance of a closed-loop feedback system, since stability and transient response are now dependent upon the sampling rate; if it is too slow, the system can be unstable since the values are not being updated rapidly enough. If we are to analyze and design feedback control systems with digital computers in the loop, we must be able to model the digital computer and associated digital-to-analog and analog-to-digital converters. The modeling of the digital computer along with associated converters is covered in the next section.

13.2 Modeling the Digital Computer

If we think about it, the form of the signals in a loop is not as important as what happens to them. For example, if analog-to-digital conversion could happen instantaneously, and time samples occurred at intervals of time that approached zero, there would be no need to differentiate between the digital signals and the analog signals. Thus, previous analysis and design techniques would be valid regardless of the presence of the digital computer.

The fact that signals are sampled at specified intervals and held causes the system performance to change with changes in sampling rate. Basically, then, the computer's effect upon the signal comes from this sampling and holding. Thus, in order to model digital control systems, we must come up with a mathematical representation of this sample-and-hold process.

Modeling the Sampler

Our objective at this point is to derive a mathematical model for the digital computer as represented by a sampler and zero-order hold. Our goal is to represent the computer as a transfer function similar to that for any subsystem. When signals are sampled, however, the Laplace transform that we have dealt with becomes a bit unwieldy. The Laplace transform can be replaced by another related transform called the *z-transform*. The *z-transform* will arise naturally from our development of the mathematical representation of the computer.

Consider the models for sampling shown in Figure 13.5. The model in Figure 13.5(a) is a switch turning on and off at a uniform sampling rate. In Figure 13.5(b), sampling can also be considered to be the product of the time waveform to be sampled, $f(t)$, and a sampling function, $s(t)$. If $s(t)$ is a sequence of pulses of width T_W, constant amplitude, and uniform rate as shown, the sampled output, $f^*_{T_W}(t)$, will consist of a sequence of sections of $f(t)$ at regular intervals. This view is equivalent to the switch model of Figure 13.5(a).

We can now write the time equation of the sampled waveform, $f^*_{T_W}(t)$. Using the model shown in Figure 13.5(b), we have

$$f^*_{T_W}(t) = f(t)s(t) = f(t) \sum_{k=-\infty}^{\infty} u(t - kT) - u(t - kT - Tw) \tag{13.1}$$

where k is an integer between $-\infty$ and $+\infty$, T is the period of the pulse train, and T_W is the pulse width.

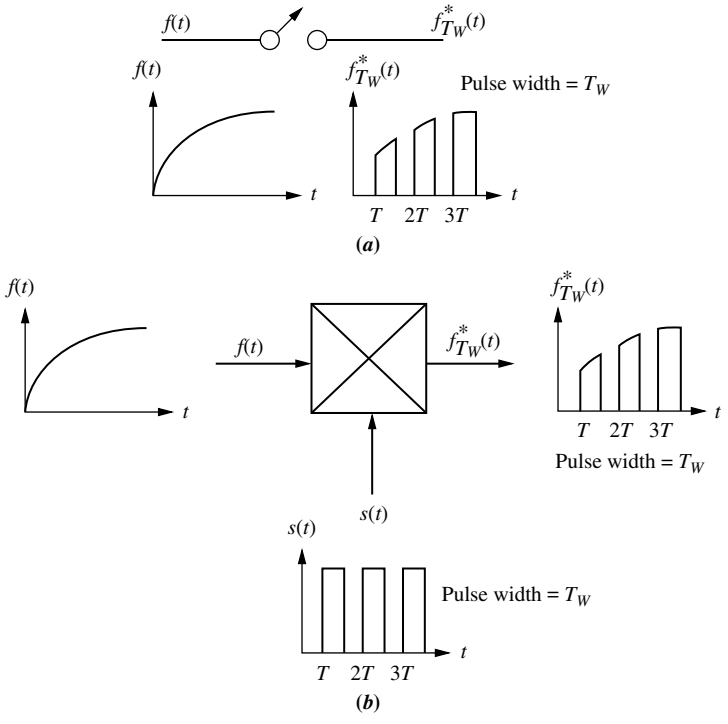


FIGURE 13.5 Two views of uniform-rate sampling: **a.** switch opening and closing; **b.** product of time waveform and sampling waveform

Since Eq. (13.1) is the product of two time functions, taking the Laplace transform in order to find a transfer function is not simple. A simplification can be made if we assume that the pulse width, T_W, is small in comparison to the period, T, such that $f(t)$ can be considered constant during the sampling interval. Over the sampling interval, then, $f(t) = f(kT)$. Hence,

$$f^*_{T_W}(t) = \sum_{k=-\infty}^{\infty} f(kT)[u(t-kT) - u(t-kT-T_W)] \tag{13.2}$$

for small T_W.

Equation (13.2) can be further simplified through insight provided by the Laplace transform. Taking the Laplace transform of Eq. (13.2), we have

$$F^*_{T_W}(s) = \sum_{k=-\infty}^{\infty} f(kT)\left[\frac{e^{-kTs}}{s} - \frac{e^{-kTs-T_Ws}}{s}\right] = \sum_{k=-\infty}^{\infty} f(kT)\left[\frac{1-e^{-T_Ws}}{s}\right]e^{-kTs} \tag{13.3}$$

Replacing e^{-T_WS} with its series expansion, we obtain

$$F^*_{T_W}(s) = \sum_{k=-\infty}^{\infty} f(kT)\left[\frac{1-\left\{1-T_Ws+\dfrac{(T_Ws)^2}{2!}-\ldots\right\}}{s}\right]e^{-kTs} \tag{13.4}$$

For small T_W, Eq. (13.4) becomes

$$F^*_{T_W}(s) = \sum_{k=-\infty}^{\infty} f(kT)\left[\frac{T_Ws}{s}\right]e^{-kTs} = \sum_{k=-\infty}^{\infty} f(kT)T_We^{-kTs} \tag{13.5}$$

Finally, converting back to the time domain, we have

$$f^*_{T_W}(t) = T_W\sum_{k=-\infty}^{\infty} f(kT)\delta(t-kT) \tag{13.6}$$

where $\delta(t-kT)$ are Dirac delta functions.

Thus, the result of sampling with rectangular pulses can be thought of as a series of delta functions whose area is the product of the rectangular pulse width and the amplitude of the sampled waveform, or $T_Wf(kT)$.

Equation (13.6) is portrayed in Figure 13.6. The sampler is divided into two parts: (1) an ideal sampler described by the portion of Eq. (13.6) that is not dependent upon the sampling waveform characteristics,

$$f^*(t)\sum_{k=-\infty}^{\infty} f(kT)\delta(t-kT) \tag{13.7}$$

and (2) the portion dependent upon the sampling waveform's characteristics, T_W.

Modeling the Zero-Order Hold

The final step in modeling the digital computer is modeling the zero-order hold that follows the sampler. Figure 13.7 summarizes the function of the zero-order hold,

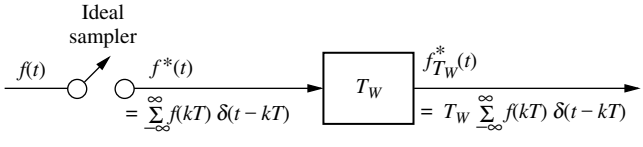


FIGURE 13.6 Model of sampling with a uniform rectangular pulse train

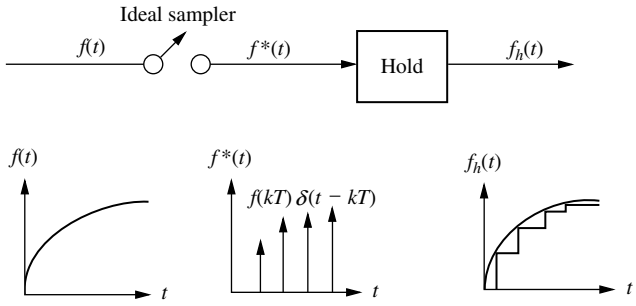


FIGURE 13.7 Ideal sampling and the zero-order hold

which is to hold the last sampled value of $f(t)$. If we assume an ideal sampler (equivalent to setting $T_W = 1$), then $f^*(t)$ is represented by a sequence of delta functions. The zero-order hold yields a staircase approximation to $f(t)$. Hence, the output from the hold is a sequence of step functions whose amplitude is $f(t)$ at the sampling instant, or $f(kT)$. We have previously seen that the transfer function of any linear system is identical to the Laplace transform of the impulse response since the Laplace transform of a unit impulse or delta function input is unity. Since a single impulse from the sampler yields a step over the sampling interval, the Laplace transform of this step, $G_h(s)$, which is the impulse response of the zero-order hold, is the transfer function of the zero-order hold. Using an impulse at zero time, the transform of the resulting step that starts at $t = 0$ and ends at $t = T$ is

$$G_h(s) = \frac{1 - e^{-Ts}}{s} \tag{13.8}$$

In a physical system, samples of the input time waveform, $f(kT)$, are held over the sampling interval. We can see from Eq. (13.8) that the hold circuit integrates the input and holds its value over the sampling interval. Since the area under the delta functions coming from the ideal sampler is $f(kT)$, we can then integrate the ideal sampled waveform and obtain the same result as for the physical system. In other words, if the ideal sampled signal, $f^*(t)$, is followed by a hold, we can use the ideal sampled waveform as the input, rather than $f^*_{T_W}(t)$.

In this section, we modeled the digital computer by cascading two elements: (1) an ideal sampler and (2) a zero-order hold. Together, the model is known as a *zero-order sample-and-hold*. The ideal sampler is modeled by Eq. (13.7), and the zero-order hold is modeled by Eq. (13.8). In the next section, we start to create a transform approach to digital systems by introducing the z-transform.

13.3 The *z*-Transform

The effect of sampling within a system is pronounced. Whereas the stability and transient response of analog systems depend upon gain and component values, sampled-data system stability and transient response also depend upon sampling rate. Our goal is to develop a transform that contains the information of sampling from which sampled-data systems can be modeled with transfer functions, analyzed, and designed with the ease and insight we enjoyed with the Laplace transform. We now develop such a transform and use the information from the last section to obtain sampled-data transfer functions for physical systems.

Equation (13.7) is the ideal sampled waveform. Taking the Laplace transform of this sampled time waveform, we obtain

$$F^*(s) = \sum_{k=0}^{\infty} f(kT)e^{-kTs} \tag{13.9}$$

Now, letting $z = e^{Ts}$, Eq. (13.9) can be written as

$$\boxed{F(z) = \sum_{k=0}^{\infty} f(kT)z^{-k}} \tag{13.10}$$

Equation (13.10) defines the *z-transform*. That is, an $F(z)$ can be transformed to $f(kT)$, or an $f(kT)$ can be transformed to $F(z)$. Alternately, we can write

$$f(kT) \Longleftrightarrow F(z) \tag{13.11}$$

Paralleling the development of the Laplace transform, we can form a table relating $f(kT)$, the value of the sampled time function at the sampling instants, to $F(z)$. Let us look at an example.

Example 13.1

z-Transform of a Time Function

PROBLEM: Find the z-transform of a sampled unit ramp.

SOLUTION: For a unit ramp, $f(kT) = kT$. Hence the ideal sampled step can be written from Eq. (13.7) as

$$f^*t = \sum_{k=0}^{\infty} kt\delta(t - kT) \tag{13.12}$$

Taking the Laplace transform, we obtain

$$F^*(s) = \sum_{k=0}^{\infty} kTe^{-kTs} \tag{13.13}$$

Converting to the z-transform by letting $e^{-kTs} = z^{-k}$, we have

$$F(z) = \sum_{k=0}^{\infty} kTz^{-k} = T\sum_{k=0}^{\infty} kz^{-k} = T(z^{-1} + 2z^{-2} + 3z^{-3} + \cdots) \tag{13.14}$$

Equation (13.14) can be converted to a closed form by forming the series for $zF(z)$ and subtracting $F(z)$. Multiplying Eq. (13.14) by z, we get

$$zF(z) = T(1 + 2z^{-1} + 3z^{-2} + \cdots) \tag{13.15}$$

Subtracting Eq. (13.14) from Eq. (13.15), we obtain

$$zF(z) - F(z) = (z - 1)F(z) = T(1 + z^{-1} + z^{-2} + \cdots) \tag{13.16}$$

But

$$\frac{1}{1 - z^{-1}} = 1 + z^{-1} + z^{-2} + z^{-3} + \cdots \tag{13.17}$$

which can be verified by performing the indicated division. Substituting Eq. (13.17) into (13.16) and solving for $F(z)$ yields

$$F(z) = T\frac{z}{(z-1)^2} \tag{13.18}$$

as the z-transform of $f(kT) = kT$.

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch13sp1 in Appendix F located at www.wiley.com/college/nise. You will learn how to find the z-transform of time functions. Example 13.1 will be solved using MATLAB and the Symbolic Math Toolbox.

The example demonstrates that any function of s, $F^*(s)$, that represents a sampled time waveform can be transformed into a function of z, $F(z)$. The final result, $F(z) = Tz/(z-1)^2$, is in a closed form, unlike $F^*(s)$. If this is the case for numerous other sampled time waveforms, then we have the convenient transform that we were looking for. In a similar way, z-transforms for other waveforms can be obtained that parallel the table of Laplace transforms in Chapter 2. A partial table of z-transforms is shown in Table 13.1, and a partial table of z-transform theorems is

TABLE 13.1 Partial table of z- and s-transforms

	$f(t)$	$F(s)$	$F(z)$	$f(kT)$
1.	$u(t)$	$\frac{1}{s}$	$\frac{z}{z-1}$	$u(kT)$
2.	t	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$	kT
3.	t^n	$\frac{n!}{s^{n+1}}$	$\lim_{a\to 0}(-1)^n\frac{d^n}{da^n}\left[\frac{z}{z-e^{-aT}}\right]$	$(kT)^n$
4.	e^{-at}	$\frac{1}{s+a}$	$\frac{z}{z-e^{-aT}}$	e^{-akT}
5.	$t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$	$(-1)^n\frac{d^n}{da^n}\left[\frac{z}{z-e^{-aT}}\right]$	$(kT)^n e^{-akT}$
6.	$\sin\omega t$	$\frac{\omega}{s^2+\omega^2}$	$\frac{z\sin\omega T}{z^2-2z\cos\omega T+1}$	$\sin\omega kT$
7.	$\cos\omega t$	$\frac{s}{s^2+\omega^2}$	$\frac{z(z-\cos\omega T)}{z^2-2z\cos\omega T+1}$	$\cos\omega kT$
8.	$e^{-at}\sin\omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$	$\frac{ze^{-aT}\sin\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$	$e^{-akT}\sin\omega kT$
9.	$e^{-at}\cos\omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$	$\frac{z^2-ze^{-aT}\cos\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$	$e^{-akT}\cos\omega kT$

TABLE 13.2 z-transform theorems

	Theorem	Name
1.	$z\{af(t)\} = aF(z)$	Linearity theorem
2.	$z\{f_1(t) + f_2(t)\} = F_1(z) + F_2(z)$	Linearity theorem
3.	$z\{e^{-aT}f(t)\} = F(e^{aT}z)$	Complex differentiation
4.	$z\{f(t - nT)\} = z^{-n}F(z)$	Real translation
5.	$z\{tf(t)\} = -Tz\dfrac{dF(z)}{dz}$	Complex differentiation
6.	$f(0) = \lim\limits_{z\to\infty} F(z)$	Initial value theorem
7.	$f(\infty) = \lim\limits_{z\to 1}(1 - z^{-1})F(z)$	Final value theorem

Note: kT may be substituted for t in the table.

shown in Table 13.2. For functions not in the table, we must perform an inverse z-transform calculation similar to the inverse Laplace transform by partial-fraction expansion. Let us now see how we can work in the reverse direction and find the time function from its z-transform.

The Inverse z-Transform

Two methods for finding the inverse z-transform (the sampled time function from its z-transform) will be described: (1) partial-fraction expansion and (2) the power series method. Regardless of the method used, remember that since the z-transform came from the sampled waveform, the inverse z-transform will yield only the values of the time function at the sampling instants. Keep this in mind as we proceed, because even as we obtain closed-form time functions as results, they are valid only at sampling instants.

Inverse z-Transforms via Partial-Fraction Expansion Recall that the Laplace transform consists of a partial fraction that yields a sum of terms leading to exponentials, that is, $A/(s + a)$. Taking this lead and looking at Table 13.1, we find that sampled exponential time functions are related to their z-transforms as follows:

$$e^{-akT} \Longleftrightarrow \frac{z}{z - e^{aT}} \tag{13.19}$$

We thus predict that a partial-fraction expansion should be of the following form:

$$F(z) = \frac{Az}{z - z_1} + \frac{Bz}{z - z_2} + \cdots \tag{13.20}$$

Since our partial-fraction expansion of $F(s)$ did not contain terms with s in the numerator of the partial fractions, we first form $F(z)/z$ to eliminate the z terms in the numerator, perform a partial-fraction expansion of $F(z)/z$, and finally multiply the result by z to replace the z's in the numerator. An example follows.

Example 13.2

Inverse z-Transform via Partial-Fraction Expansion

PROBLEM: Given the function in Eq. (13.21), find the sampled time function.

$$F(z) = \frac{0.5z}{(z - 0.5)(z - 0.7)} \tag{13.21}$$

SOLUTION: Begin by dividing Eq. (13.21) by z and performing a partial-fraction expansion.

$$\frac{F(z)}{z} = \frac{0.5}{(z-0.5)(z-0.7)} = \frac{A}{z-0.5} + \frac{B}{z-0.7} = \frac{-2.5}{z-0.5} + \frac{2.5}{z-0.7} \tag{13.22}$$

Next, multiply through by z.

$$F(z) = \frac{0.5z}{(z-0.5)(z-0.7)} = \frac{-2.5z}{z-0.5} + \frac{2.5z}{z-0.7} \tag{13.23}$$

Using Table 13.1, we find the inverse z-transform of each partial fraction. Hence, the value of the time function at the sampling instants is

$$f(kT) = -2.5(0.5)^k + 2.5(0.7)^k \tag{13.24}$$

Also, from Eqs. (13.7) and (13.24), the ideal sampled time function is

$$f^*(t) = \sum_{k=-\infty}^{\infty} f(kT)\delta(t-kT) = \sum_{k=-\infty}^{\infty} [-2.5(0.5)^k + 2.5(0.7)^k]\delta(t-kT) \tag{13.25}$$

If we substitute $k = 0, 1, 2$, and 3, we can find the first four samples of the ideal sampled time waveform. Hence,

$$f^*(t) = 0\delta(t) + 0.5\delta(t-T) + 0.6\delta(t-2T) + 0.545\delta(t-3T) \tag{13.26}$$

Symbolic Math
SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch13sp2 in Appendix F located at www.wiley.com/college/nise. You will learn how to find the inverse z-transform of sampled time functions. Example 13.2 will be solved using MATLAB and the Symbolic Math Toolbox.

Inverse z-Transform via the Power Series Method The values of the sampled time waveform can also be found directly from $F(z)$. Although this method does not yield closed-form expressions for $f(kT)$, it can be used for plotting. The method consists of performing the indicated division, which yields a power series for $F(z)$. The power series can then be easily transformed into $F^*(s)$ and $f^*(t)$.

Example 13.3

Inverse z-Transform via Power Series

PROBLEM: Given the function in Eq. (13.21), find the sampled time function.

SOLUTION: Begin by converting the numerator and denominator of $F(z)$ to polynomials in z.

$$F(z) = \frac{0.5z}{(z-0.5)(z-0.7)} = \frac{0.5z}{z^2 - 1.2z + 0.35} \tag{13.27}$$

Now perform the indicated division.

$$\begin{array}{r} 0.5z^{-1} + 0.6z^{-2} + 0.545z^{-3} \\ z^2 - 1.2z + 0.35 \overline{)\,0.5z} \\ \underline{0.5z - 0.6 + 0.175z^{-1}} \\ 0.6 - 0.175z^{-1} \\ \underline{0.6 - 0.720z^{-1} + 0.21} \\ 0.545z^{-1} - 0.21 \end{array} \tag{13.28}$$

Using the numerator and the definition of z, we obtain

$$F^*(s) = 0.5e^{-Ts} + 0.6e^{-2Ts} + 0.545e^{-3Ts} + \cdots \tag{13.29}$$

from which

$$f^*(t) = 0.5\delta(t - T) + 0.6\delta(t - 2T) + 0.545\delta(t - 3T) + \cdots \tag{13.30}$$

You should compare Eq. (13.30) with Eq. (13.26), the result obtained via partial expansion.

Skill-Assessment Exercise 13.1

PROBLEM: Derive the z-transform for $f(t) = \sin \omega t\, u(t)$.

ANSWER: $F(z) = \dfrac{z^{-1}\sin(\omega T)}{1 - 2z^{-1}\cos(\omega T) + z^{-2}}$

WileyPLUS WPCS Control Solutions

The complete solution is located at www.wiley.com/college/nise.

Skill-Assessment Exercise 13.2

PROBLEM: Find $f(kT)$ if $F(z) = \dfrac{z(z+1)(z+2)}{(z-0.5)(z-0.7)(z-0.9)}$.

ANSWER: $f(kT) = 46.875(0.5)^k - 114.75(0.7)^k + 68.875(0.9)^k$

The complete solution is located at www.wiley.com/college/nise.

13.4 Transfer Functions

Now that we have established the z-transform, let us apply it to physical systems by finding transfer functions of sampled-data systems. Consider the continuous system shown in Figure 13.8 (a). If the input is sampled as shown in Figure 13.8(b), the output is still a continuous signal. If, however, we are satisfied with finding the output at the sampling instants and not in between, the representation of the sampled-data system

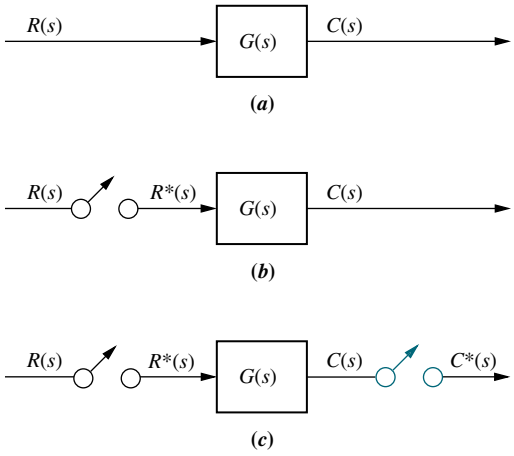


FIGURE 13.8 Sampled-data systems: **a.** continuous; **b.** sampled input; **c.** sampled input and output

Note: Phantom sampler is shown in color.

can be greatly simplified. Our assumption is visually described in Figure 13.8(c), where the output is conceptually sampled in synchronization with the input by a phantom sampler. Using the concept described in Figure 13.8(c), we derive the pulse transfer function of $G(s)$.

Derivation of the Pulse Transfer Function

Using Eq. (13.7), we find that the sampled input, $r^*(t)$, to the system of Figure 13.8(c) is

$$r^*(t) = \sum_{n=0}^{\infty} r(nT)\delta(t - nT) \tag{13.31}$$

which is a sum of impulses. Since the impulse response of a system, $G(s)$, is $g(t)$, we can write the time output of $G(s)$ as the sum of impulse responses generated by the input, Eq. (13.31). Thus,

$$c(t) = \sum_{n=0}^{\infty} r(nT)g(t - nT) \tag{13.32}$$

From Eq. (13.10),

$$C(z) = \sum_{k=0}^{\infty} c(kT)z^{-k} \tag{13.33}$$

Using Eq. (13.32) with $t = kT$, we obtain

$$c(kT) = \sum_{n=0}^{\infty} r(nT)g(kT - nT) \tag{13.34}$$

Substituting Eq. (13.34) into Eq. (13.33), we obtain

$$C(z) = \sum_{k=0}^{\infty}\sum_{n=0}^{\infty} r(nT)g[(k - n)T]z^{-k} \tag{13.35}$$

Letting $m = k - n$, we find

$$\begin{aligned} C(z) &= \sum_{m+n=0}^{\infty} \sum_{n=0}^{\infty} r(nT)g(mT)z^{-(m+n)} \\ &= \left\{\sum_{m=0}^{\infty} g(mT)z^{-m}\right\}\left\{\sum_{n=0}^{\infty} r(nT)z^{-n}\right\} \end{aligned} \tag{13.36}$$

where the lower limit, $m+n$, was changed to m. The reasoning is that $m+n=0$ yields negative values of m for all $n>0$. But, since $g(mT)=0$ for all $m<0$, m is not less than zero. Alternately, $g(t)=0$ for $t<0$. Thus, $n=0$ in the first sum's lower limit.

Using the definition of the z-transform, Eq. (13.36) becomes

$$C(z) = \sum_{m=0}^{\infty} g(mT)z^{-m} \sum_{n=0}^{\infty} r(nT)z^{-n} = G(z)R(z) \tag{13.37}$$

Equation (13.37) is a very important result, since it shows that the transform of the sampled output is the product of the transforms of the sampled input and the pulse transfer function of the system. Remember that although the output of the system is a continuous function, we had to make an assumption of a sampled output (phantom sampler) in order to arrive at the compact result of Eq. (13.37).

One way of finding the pulse transfer function, $G(z)$, is to start with $G(s)$, find $g(t)$, and then use Table 13.1 to find $G(z)$. Let us look at an example.

Example 13.4

Converting $G_1(s)$ in Cascade with z.o.h. to $G(z)$

PROBLEM: Given a z.o.h. in cascade with $G_1(s) = (s+2)/(s+1)$ or

$$G(s) = \frac{1-e^{-Ts}}{s}\frac{(s+2)}{(s+1)} \tag{13.38}$$

find the sampled-data transfer function, $G(z)$, if the sampling time, T, is 0.5 second.

SOLUTION: Equation (13.38) represents a common occurrence in digital control systems, namely a transfer function in cascade with a zero-order hold. Specifically, $G_1(s) = (s+2)/(s+1)$ is in cascade with a zero-order hold, $(1-e^{-Ts})/s$. We can formulate a general solution to this type of problem by moving the s in the denominator of the zero-order hold to $G_1(s)$, yielding

$$G(s) = (1-e^{-Ts})\frac{G_1(s)}{s} \tag{13.39}$$

from which

$$G(z) = (1-z^{-1})z\left\{\frac{G_1(s)}{s}\right\} = \frac{z-1}{z}z\left\{\frac{G_1(s)}{s}\right\} \tag{13.40}$$

Thus, begin the solution by finding the impulse response (inverse Laplace transform) of $G_1(s)/s$. Hence,

$$G_2(s) = \frac{G_1(s)}{s} = \frac{s+2}{s(s+1)} = \frac{A}{s} + \frac{B}{s+1} = \frac{2}{s} - \frac{1}{s+1} \tag{13.41}$$

TryIt 13.1

Use MATLAB, the Control System Toolbox, and the following statements to find $G_1(s)$ in Example 13.4 given $G(z)$ in Eq. (13.46)

```
num=0.213;
den=0.607;
k=1;
T=0.5;
Gz=zpk(num,den,K,T)
Gs=d2c(Gz,'zoh')
```

Taking the inverse Laplace transform, we get

$$g_2(t) = 2 - e^{-t} \tag{13.42}$$

from which

$$g_2(kT) = 2 - e^{-kt} \tag{13.43}$$

Using Table 13.1, we find

$$G_2(z) = \frac{2z}{z-1} - \frac{z}{z-e^{-T}} \tag{13.44}$$

Substituting $T = 0.5$ yields

$$G_2(z) = z\left\{\frac{G_1(s)}{s}\right\} = \frac{2z}{z-1} - \frac{z}{z-0.607} = \frac{z^2 - 0.213z}{(z-1)(z-0.607)} \tag{13.45}$$

From Eq. (13.40),

$$G(z) = \frac{z-1}{z} G_2(z) = \frac{z-0.213}{z-0.607} \tag{13.46}$$

MATLAB ML

Students who are using MATLAB should now run ch13p1 in Appendix B. You will learn how to use MATLAB to convert $G_1(s)$ in cascade with a zero-order hold to $G(z)$. This exercise solves Example 13.4 using MATLAB.

Symbolic Math SM

Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Symbolic Math Toolbox should now run ch13sp3 in Appendix F located at www.wiley.com/college/nise. MATLAB's Symbolic Math Toolbox yields an alternative method of finding the z-transform of a transfer function in cascade with a zero-order hold. Example 13.4 will be solved using MATLAB and the Symbolic Math Toolbox with a method that follows closely the hand calculation shown in that example.

MATLAB ML

Students who are using MATLAB should now run ch13p2 in Appendix B. You will learn how to use MATLAB to convert $G(s)$ to $G(z)$ when $G(s)$ is not in cascade with a zero-order hold. This is the same as finding the z-transform of $G(s)$.

MATLAB ML

Students who are using MATLAB should now run ch13p3 in Appendix B. You will learn how to create digital transfer functions directly.

MATLAB ML

Students who are using MATLAB should now run ch13p4 in Appendix B. You will learn how to use MATLAB to convert $G(z)$ to $G(s)$ when $G(s)$ is not in cascade with a zero-order hold. This is the same as finding the Laplace transform of $G(z)$.

Skill-Assessment Exercise 13.3

TryIt 13.2

Use MATLAB, the Control System Toolbox, and the following statements to solve Skill-Assessment Exercise 13.3.

```
Gs=zpk([],-4,8)
Gz=c2d(Gs,0.25,'zoh')
```

PROBLEM: Find $G(z)$ for $G(s) = 8/(s+4)$ in cascade with a zero-order sample and hold. The sampling period is 0.25 second.

WileyPLUS WPCS Control Solutions

ANSWER: $G(z) = 1.264/(z - 0.3679)$

The complete solution is located at www.wiley.com/college/nise.

The major discovery in this section is that once the pulse transfer function, $G(z)$, of a system is obtained, the transform of the sampled output response, $C(z)$, for a given sampled input can be evaluated using the relationship $C(z) = R(z)G(z)$. Finally, the time function can be found by taking the inverse z-transform, as covered in Section 13.3. In the next section, we look at block diagram reduction for digital systems.

13.5 Block Diagram Reduction

Up to this point, we have defined the z-transform and the sampled-data system transfer function and have shown how to obtain the sampled response. Basically, we are paralleling our discussions of the Laplace transform in Chapters 2 and 4. We now draw a parallel with some of the objectives of Chapter 5, namely block diagram reduction. Our objective here is to be able to find the closed-loop sampled-data transfer function of an arrangement of subsystems that have a computer in the loop.

When manipulating block diagrams for sampled-data systems, you must be careful to remember the definition of the sampled-data system transfer function (derived in the last section) to avoid mistakes. For example, $z\{G_1(s)G_2(s)\} \neq G_1(z)G_2(z)$, where $z\{G_1(s)G_2(s)\}$ denotes the z-transform. The s-domain functions have to be multiplied together before taking the z-transform. In the ensuing discussion, we use the notation $G_1G_2(s)$ to denote a single function that is $G_1(s)G_2(s)$ after evaluating the product. Hence, $z\{G_1(s)G_2(s)\} = z\{G_1G_2(s)\} = G_1G_2(z) \neq G_1(z)G_2(z)$.

Let us look at the sampled-data systems shown in Figure 13.9. The sampled-data systems are shown under the column marked s. Their z-transforms are shown under the column marked z. The standard system that we derived earlier is shown in Figure 13.9(a), where the transform of the output, $C(z)$, is equal to $R(z)G(z)$. This system forms the basis for the other entries in Figure 13.9.

In Figure 13.9(b), there is no sampler between $G_1(s)$ and $G_2(s)$. Thus, we can think of a single function, $G_1(s)G_2(s)$, denoted $G_1G_2(s)$, existing between the two samplers and yielding a single transfer function, as shown in Figure 13.9(a). Hence, the pulse transfer function is $z\{G_1G_2(s)\} = G_1G_2(z)$. The transform of the output, $C(z) = R(z)G_1G_2(z)$.

In Figure 13.9(c), we have the cascaded two subsystems of the type shown in Figure 13.9(a). For this case, then, the z-transform is the product of the two z-transforms, or $G_2(z)G1(z)$. Hence the transform of the output $C(z) = R(z)G_2(z)G_1(z)$.

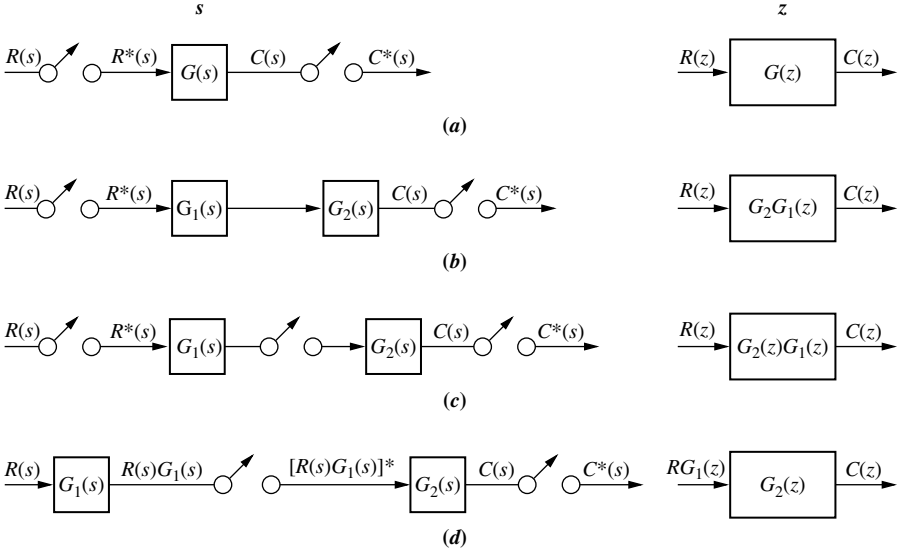


FIGURE 13.9 Sampled-data systems and their z-transforms

Finally, in Figure 13.9(d), we see that the continuous signal entering the sampler is $R(s)G_1(s)$. Thus, the model is the same as Figure 13.9(a) with $R(s)$ replaced by $R(s)G_1(s)$, and $G_2(s)$ in Figure 13.9(d) replacing $G(s)$ in Figure 13.9(a). The z-transform of the input to $G_2(s)$ is $z\{R(s)G_1(s)\} = z\{RG_1(s)\} = RG_1(z)$. The pulse transfer function for the system $G_2(s)$ is $G_2(z)$. Hence, the output $C(z) = RG_1(z)G_2(z)$.

Using the basic forms shown in Figure 13.9, we can now find the z-transform of feedback control systems. We have shown that any system, $G(s)$, with sampled input and sampled output, such as that shown in Figure 13.9(a), can be represented as a sampled-data transfer function, $G(z)$. Thus, we want to perform block diagram manipulations that result in subsystems, as well as the entire feedback system, that have sampled inputs and sampled outputs. Then we can make the transformation to sampled-data transfer functions. An example follows.

Example 13.5

Pulse Transfer Function of a Feedback System

PROBLEM: Find the z-transform of the system shown in Figure 13.10(a).

SOLUTION: The objective of the problem is to proceed in an orderly fashion, starting with the block diagram of Figure 13.10(a) and reducing it to the one shown in Figure 13.10(f).

One operation we can always perform is to place a phantom sampler at the output of any subsystem that has a sampled input, provided that the nature of the signal sent to any other subsystem is not changed. For example in Figure 13.10(b), phantom sampler $S4$ can be added. The justification for this, of course, is that the

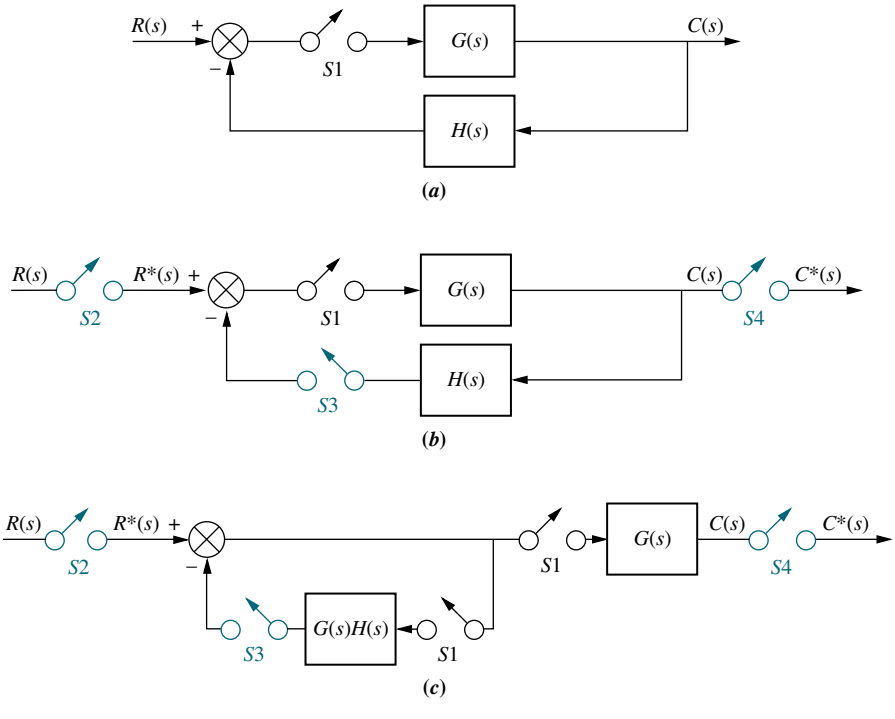
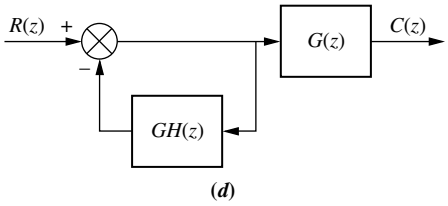
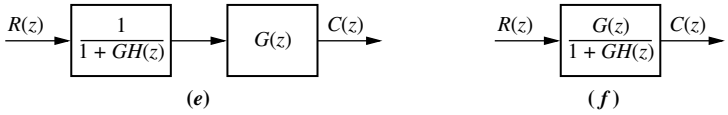


FIGURE 13.10 Steps in block diagram reduction of a sampled-data system (*figure continues*)

Note: Phantom samplers are shown in color.



(*d*)



(*e*) (*f*)

FIGURE 13.10 (*Continued*)

output of a sampled-data system can only be found at the sampling instants anyway, and the signal is not an input to any other block.

Another operation that can be performed is to add phantom samplers $S2$ and $S3$ at the input to a summing junction whose output is sampled. The justification for this operation is that the sampled sum is equivalent to the sum of the sampled inputs, provided, of course, that all samplers are synchronized.

Next, move sampler $S1$ and $G(s)$ to the right past the pickoff point, as shown in Figure 13.10(c). The motivation for this move is to yield a sampler at the input of $G(s)H(s)$ to match Figure 13.9(b). Also, $G(s)$ with sampler $S1$ at the input and sampler $S4$ at the output matches Figure 13.9(a). The closed-loop system now has a sampled input and a sampled output.

$G(s)H(s)$ with samplers $S1$ and $S3$ becomes $GH(z)$, and $G(s)$ with samplers $S1$ and $S4$ becomes $G(z)$, as shown in Figure 13.10(d). Also, converting $R^*(s)$ to $R(z)$ and $C^*(s)$ to $C(z)$, we now have the system represented totally in the z-domain.

The equations derived in Chapter 5 for transfer functions represented with the Laplace transform can be used for sampled-data transfer functions with only a change in variables from s to z. Thus, using the feedback formula, we obtain the first block of Figure 13.10(e). Finally, multiplication of the cascaded sampled-data systems yields the final result shown in Figure 13.10(f).

Skill-Assessment Exercise 13.4

PROBLEM: Find $T(z) = C(z)/R(z)$ for the system shown in Figure 13.11.

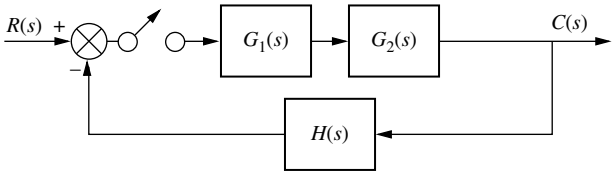


FIGURE 13.11 Digital system for Skill-Assessment Exercise 13.4

ANSWER: $T(z) = \dfrac{G_1G_2(z)}{1 + HG_1G_2(z)}$

The complete solution is located at www.wiley.com/college/nise.

This section paralleled Chapter 5 by showing how to obtain the closed-loop, sampled-data transfer function for a collection of subsystems. The next section parallels the discussion of stability in Chapter 6.

13.6 Stability

The glaring difference between analog feedback control systems and digital feedback control systems, such as the one shown in Figure 13.12, is the effect that the sampling rate has on the transient response. Changes in sampling rate not only change the nature of the response from overdamped to underdamped, but also can turn a stable system into an unstable one. As we proceed with our discussion, these effects will become apparent. You are encouraged to be on the lookout.

We now discuss the stability of digital systems from two perspectives: (1) z-plane and (2) s-plane. We will see that the Routh-Hurwitz criterion can be used only if we perform our analysis and design on the s-plane.

Digital System Stability via the z-Plane

In the s-plane, the region of stability is the left half-plane. If the transfer function, $G(s)$, is transformed into a sampled-data transfer function, $G(z)$, the region of stability on the z-plane can be evaluated from the definition, $z = e^{Ts}$. Letting $s = \alpha + j\omega$, we obtain

$$\begin{aligned} z = e^{Ts} &= e^{T(\alpha+j\omega)} = e^{\alpha T} e^{j\omega T} \\ &= e^{\alpha T}(\cos \omega T + j \sin \omega T) \\ &= e^{\alpha T} \angle\, \omega T \end{aligned} \tag{13.47}$$

since $(\cos \omega T + j \sin \omega T) = 1 \angle\, \omega T$.

Each region of the s-plane can be mapped into a corresponding region on the z-plane (see Figure 13.13). Points that have positive values of α are in the right half



FIGURE 13.12 A lathe using digital numerical control (© David J. Green—Industry/Alamy)

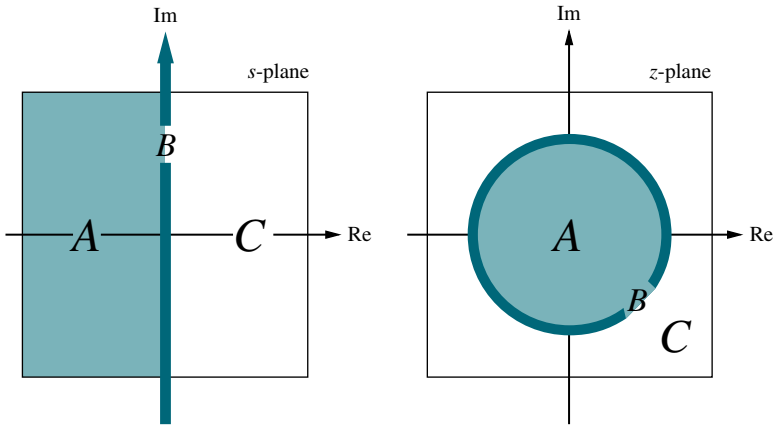


FIGURE 13.13 Mapping regions of the s-plane onto the z-plane

of the s-plane, region C. From Eq. (13.47), the magnitudes of the mapped points are $e^{\alpha T} > 1$. Thus points in the right half of the s-plane map into points outside the unit circle on the z-plane.

Points on the $j\omega$-axis, region B, have zero values of α and yield points on the z-plane with magnitude $= 1$, the unit circle. Hence, points on the $j\omega$-axis in the s-plane map into points on the unit circle on the z-plane.

Finally, points on the s-plane that yield negative values of a α(left–half-plane roots, region A) map into the inside of the unit circle on the z-plane.

Thus, a digital control system is (1) stable if all poles of the closed-loop transfer function, $T(z)$, are inside the unit circle on the z-plane, (2) unstable if any pole is outside the unit circle and/or there are poles of multiplicity greater than one on the unit circle, and (3) marginally stable if poles of multiplicity one are on the unit circle and all other poles are inside the unit circle. Let us look at an example.

Example 13.6

Modeling and Stability

PROBLEM: The missile shown in Figure 13.14(a) can be aerodynamically controlled by torques created by the deflection of control surfaces on the missile's body. The commands to deflect these control surfaces come from a computer that uses tracking data along with programmed guidance equations to determine whether the missile is on track. The information from the guidance equations is used to develop flight-control commands for the missile. A simplified model is shown in Figure 13.14(b). Here the computer performs the function of controller by using tracking information to develop input commands to the missile. An accelerometer in the missile detects the actual acceleration, which is fed back to the computer. Find the closed-loop digital transfer function for this system and determine if the system is stable for $K = 20$ and $K = 100$ with a sampling interval of $T = 0.1$ second.

SOLUTION: The input to the control system is an acceleration command developed by the computer. The computer can be modeled by a sample-and-hold. The s-plane model is shown in Figure 13.14(c). The first step in finding the z-plane model is to find $G(z)$, the forward-path transfer function. From Figure 13.14(c) or (d),

$$G(s) = \frac{1 - e^{-Ts}}{s} \frac{Ka}{s(s + a)} \tag{13.48}$$

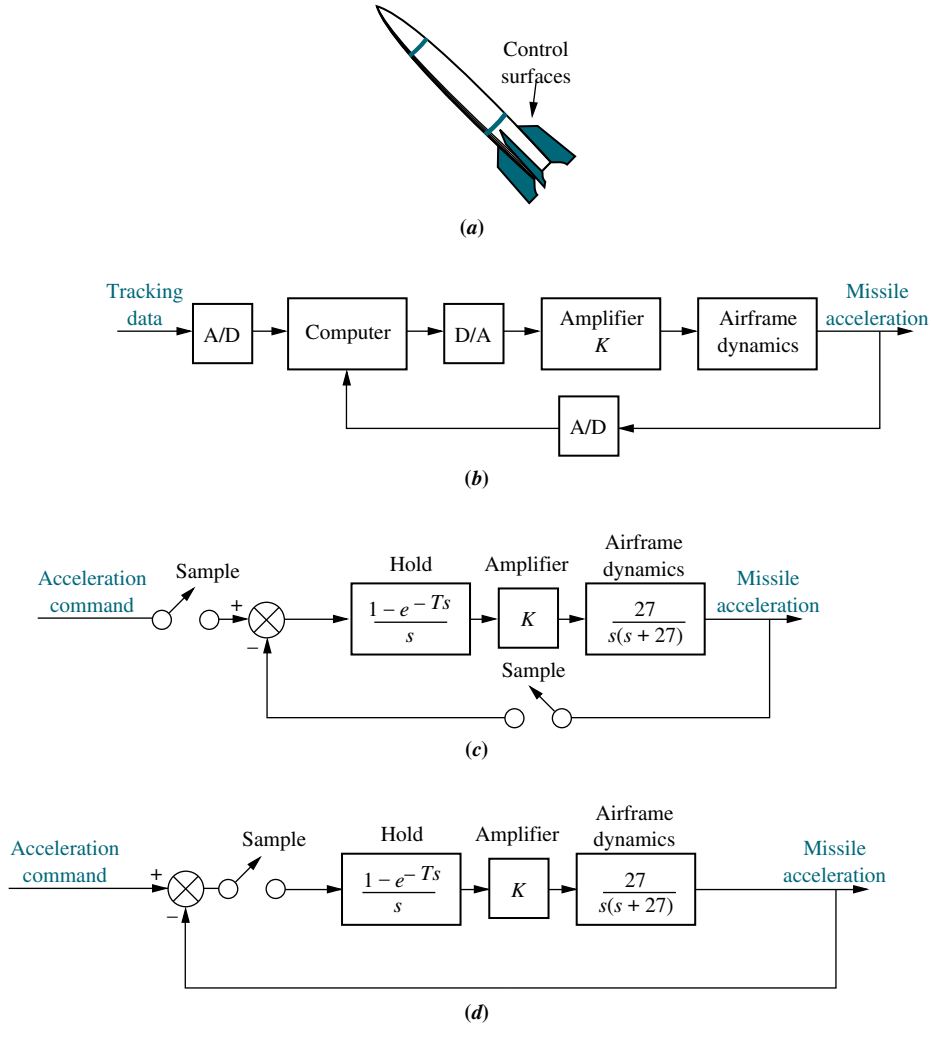


FIGURE 13.14 Finding stability of a missile control system: **a.** missile; **b.** conceptual block diagram; **c.** block diagram; **d.** block diagram with equivalent single sampler

where $a = 27$. The z-transform, $G(z)$, is $(1 - z^{-1})z\{Ka/[s^2(s + a)]\}$.

The term $Ka/[s^2(s + a)]$ is first expanded by partial fractions, after which we find the z-transform of each term from Table 13.1. Hence,

$$\begin{aligned} z\left\{\frac{Ka}{s^2(s+a)}\right\} &= Kz\left\{\frac{a}{s^2(s+a)}\right\} = Kz\left\{\frac{1}{s^2} - \frac{1/a}{s} + \frac{1/a}{s+a}\right\} \\ &= K\left\{\frac{Tz}{(z-1)^2} - \frac{z/a}{z-1} + \frac{z/a}{z-e^{-aT}}\right\} \\ &= K\left\{\frac{Tz}{(z-1)^2} - \frac{(1-e^{-aT})z}{a(z-1)(z-e^{-aT})}\right\} \end{aligned} \tag{13.49}$$

Thus,

$$G(z) = K\left\{\frac{T(z-e^{-aT}) - (z-1)\left(\dfrac{1-e^{-aT}}{a}\right)}{(z-1)(z-e^{-aT})}\right\} \tag{13.50}$$

Letting $T = 0.1$ and $a = 27$, we have

$$G(z) = \frac{K(0.0655z + 0.02783)}{(z-1)(z-0.0672)} \tag{13.51}$$

Finally, we find the closed-loop transfer function, $T(z)$, for a unity feedback system:

$$T(z) = \frac{G(z)}{1+G(z)} = \frac{K(0.0655z + 0.02783)}{z^2 + (0.0655K - 1.0672)z + (0.02783K + 0.0672)} \tag{13.52}$$

The stability of the system is found by finding the roots of the denominator. For $K = 20$, the roots of the denominator are $0.12 \pm j0.78$. The system is thus stable for $K = 20$, since the poles are inside the unit circle. For $K = 100$, the poles are at -0.58 and -4.9. Since one of the poles is outside the unit circle, the system is unstable for $K = 100$.

MATLAB

ML

`Students who are using MATLAB should now run ch13p5 in Appendix B. You will learn how to use MATLAB to determine the range of K for stability in a digital system. This exercise solves Example 13.6 using MATLAB.`

In the case of continuous systems, the determination of stability hinges upon our ability to determine whether the roots of the denominator of the closed-loop transfer function are in the stable region of the s-plane. The problem for high-order systems is complicated by the fact that the closed-loop transfer function denominator is in polynomial form, not factored form. The same problem surfaces with closed-loop sampled-data transfer functions.

Tabular methods for determining stability, such as the Routh-Hurwitz method used for higher-order continuous systems, exist for sampled-data systems. These methods, which are not covered in this introductory chapter to digital control systems, can be used to determine stability in higher-order digital systems. If you wish to go further into the area of digital system stability, you are encouraged to look at Raible's tabular method or Jury's stability test for determining the number of a sampled-data system's closed-loop poles that exist outside the unit circle and thus indicate instability.[2]

The following example demonstrates the effect of sampling rate on the stability of a closed-loop feedback control system. All parameters are constant except for the sampling interval, T. We will see that varying T will lead us through regions of stability and instability just as though we were varying the forward-path gain, K.

Example 13.7

Range of *T* for Stability

PROBLEM: Determine the range of sampling interval, T, that will make the system shown in Figure 13.15 stable, and the range that will make it unstable.

SOLUTION: Since $H(s) = 1$, the z-transform of the closed-loop system, $T(z)$, is found from Figure 13.10 to be

$$T(z) = \frac{G(z)}{1+G(z)} \tag{13.53}$$

[2] A discussion of Raible's tabular method and Jury's stability test can be found in *Kuo (1980: 278–286)*.

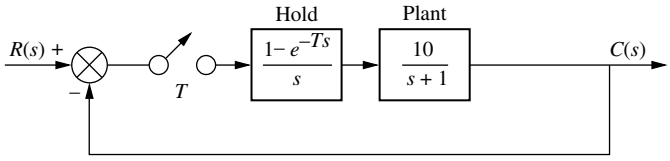


FIGURE 13.15 Digital system for Example 13.7

To find $G(z)$, first find the partial-fraction expansion of $G(s)$.

$$G(s) = 10\frac{1 - e^{-Ts}}{s(s+1)} = 10(1 - e^{-Ts})\left(\frac{1}{s} - \frac{1}{s+1}\right) \tag{13.54}$$

Taking the z-transform, we obtain

$$G(z) = \frac{10(z-1)}{z}\left[\frac{z}{z-1} - \frac{z}{z - e^{-T}}\right] = 10\frac{(1 - e^{-T})}{(z - e^{-T})} \tag{13.55}$$

Substituting Eq. (13.55) into (13.53) yields

$$T(z) = \frac{10(1 - e^{-T})}{z - (11e^{-T} - 10)} \tag{13.56}$$

The pole of Eq. (13.56), $(11e^{-T} - 10)$, monotonically decreases from $+1$ to -1 for $0 < T < 0.2$. For $0.2 < T < \infty$, $(11e^{-T} - 10)$ monotonically decreases from -1 to -10. Thus, the pole of $T(z)$ will be inside the unit circle, and the system will be stable if $0 < T < 0.2$. In terms of frequency, where $f = 1/T$, the system will be stable as long as the sampling frequency is $1/0.2 = 5$ hertz or greater.

We now have found, via the z-plane, that sampled systems are stable if their poles are inside the unit circle. Unfortunately, this stability criterion precludes the use of the Routh-Hurwitz criterion, which detects roots in the right half-plane rather than outside the unit circle. However, another method exists that allows us to use the familiar s-plane and the Routh-Hurwitz criterion to determine the stability of a sampled system. Let us introduce this topic.

Bilinear Transformations

Bilinear transformations give us the ability to apply our s-plane analysis and design techniques to digital systems. We can analyze and design on the s-plane as we have done in Chapters 8 and 9 and then, using these transformations, convert the results to a digital system that contains the same properties. Let us look further into this topic.

We can consider $z = e^{Ts}$ and its inverse, $s = (1/T)\ln z$, as the exact transformations between z and s. Thus, if we have $G(z)$ and substitute $z = e^{Ts}$, we obtain $G(e^{Ts})$ as the result of converting to s. Similarly, if we have $G(s)$ and substitute $s = (1/T)\ln z$, we obtain $G((1/T)\ln z)$ as the result of converting to z. Unfortunately, both transformations yield transcendental functions, which we of course take care of through the rather complicated z-transform.

What we would like is a simple transformation that would yield linear arguments when transforming in both directions (bilinear) through direct substitution and without the complicated z-transform.

Bilinear transformations of the form

$$z = \frac{as + b}{cs + d} \tag{13.57}$$

and its inverse,

$$s = \frac{-dz + b}{cz - a} \tag{13.58}$$

have been derived to yield linear variables in s and z. Different values of a, b, c, and d have been derived for particular applications and yield various degrees of accuracy when comparing properties of the continuous and sampled functions.

For example, in the next subsection we will see that a particular choice of coefficients will take points on the unit circle and map them into points on the $j\omega$-axis. Points outside the unit circle will be mapped into the right half-plane, and points inside the unit circle will be mapped into the left half-plane. Thus, we will be able to make a simple transformation from the z-plane to the s-plane and obtain stability information about the digital system by working in the s-plane.

Since the transformations are not exact, only the property for which they are designed can be relied upon. For the stability transformation just discussed, we cannot expect the resulting $G(s)$ to have the same transient response as $G(z)$. Another transformation will be covered that will retain that property.

Digital System Stability via the *s*-Plane

In this subsection, we look at a bilinear transformation that maps $j\omega$-axis points on the s-plane to unit-circle points on the z-plane. Further, the transformation maps right–half-plane points on the s-plane to points outside the unit circle on the z-plane. Finally, the transformation maps left–half-plane points on the s-plane to points inside the unit circle on the z-plane. Thus, we are able to transform the denominator of the pulsed transfer function, $D(z)$, to the denominator of a continuous transfer function, $D(s)$, and use the Routh-Hurwitz criterion to determine stability.

The bilinear transformation

$$\boxed{s = \frac{z+1}{z-1}} \tag{13.59}$$

and its inverse

$$\boxed{z = \frac{s+1}{s-1}} \tag{13.60}$$

perform the required transformation (*Kuo*, *1995*). We can show this fact as follows: Letting $s = \alpha + j\omega$ and substituting into Eq. (13.60),

$$z = \frac{(\alpha + 1) + j\omega}{(\alpha - 1) + j\omega} \tag{13.61}$$

from which

$$|z| = \frac{\sqrt{(\alpha+1)^2 + \omega^2}}{\sqrt{(\alpha-1)^2 + \omega^2}} \tag{13.62}$$

Thus,

$$|z| < 1 \quad \text{when } \alpha < 0 \tag{13.63a}$$

$$|z| > 1 \quad \text{when } \alpha > 0 \tag{13.63b}$$

and

$$|z| = 1 \quad \text{when } \alpha = 0 \tag{13.63c}$$

Let us look at an example that shows how the stability of sampled systems can be found using this bilinear transformation and the Routh-Hurwit criterion.

Example 13.8

Stability via Routh-Hurwitz

PROBLEM: Given $T(z) = N(z)/D(z)$, where $D(z) = z^3 - z^2 - 0.2z + 0.1$, use the Routh-Hurwitz criterion to find the number of z-plane poles of $T(z)$ inside, outside, and on the unit circle. Is the system stable?

SOLUTION: Substitute Eq. (13.60) into $D(z) = 0$ and obtain[3]

$$s^3 - 19s^2 - 45s - 17 = 0 \tag{13.64}$$

The Routh table for Eq. (13.64), Table 13.3, shows one root in the right–half-plane and two roots in the left–half-plane. Hence, $T(z)$ has one pole outside the unit circle, no poles on the unit circle, and two poles inside the unit circle. The system is unstable because of the pole outside the unit circle.

TABLE 13.3 Routh table for Example 13.8

s^3	1	−45
s^2	19	−17
s^1	−45.89	0
s^0	−17	0

Skill-Assessment Exercise 13.5

WileyPLUS
WPCS
Control Solutions

PROBLEM: Determine the range of sampling interval, T, that will make the system shown in Figure 13.16 stable.

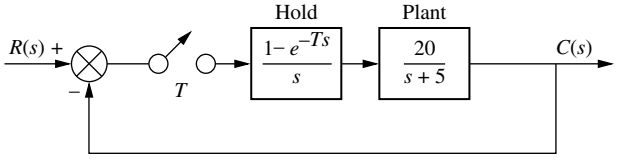


FIGURE 13.16 Digital system for Skill-Assessment Exercise 13.5

ANSWER: $0 < T < 0.1022$ second

The complete solution is located at www.wiley.com/college/nise.

[3] Symbolic math software, such as MATLAB's Symbolic Math Toolbox, is recommended to reduce the labor required to perform the transformation.

Skill-Assessment Exercise 13.6

PROBLEM: Given $T(z) = N(z)/D(z)$, where $D(z) = z^3 - z^2 - 0.5z + 0.3$, use the Routh-Hurwitz criterion to find the number of z-plane poles of $T(z)$ inside, outside, and on the unit circle. Is the system stable?

ANSWER: $T(z)$ has one pole outside the unit circle, no poles on the unit circle, and two poles inside the unit circle. The system is unstable.

The complete solution is located at www.wiley.com/college/nise.

In this section, we covered the concepts of stability for digital systems. Both z- and s-plane perspectives were discussed. Using a bilinear transformation, we are able to use the Routh-Hurwitz criterion to determine stability.

The highlight of the section is that sampling rate (along with system parameters, such as gain and component values) helps to determine or destroy the stability of a digital system. In general, if the sampling rate is too slow, the closed-loop digital system will be unstable. We now move from stability to steady-state errors, paralleling our previous discussion of steady-state errors in analog systems.

13.7 Steady-State Errors

We now examine the effect of sampling upon the steady-state error for digital systems. Any general conclusion about the steady-state error is difficult because of the dependence of those conclusions upon the placement of the sampler in the loop. Remember that the position of the sampler could change the open-loop transfer function. In the discussion of analog systems, there was only one open-loop transfer function, $G(s)$, upon which the general theory of steady-state error was based and from which came the standard definitions of static error constants. For digital systems, however, the placement of the sampler changes the open-loop transfer function and thus precludes any general conclusions. In this section, we assume the typical placement of the sampler after the error and in the position of the cascade controller, and we derive our conclusions accordingly about the steady-state error of digital systems.

Consider the digital system in Figure 13.17(a), where the digital computer is represented by the sampler and zero-order hold. The transfer function of the plant is represented by $G_1(s)$ and the transfer function of the z.o.h. by $(1 - e^{-Ts})/s$. Letting $G(s)$ equal the product of the z.o.h. and $G_1(s)$, and using the block diagram reduction techniques for sampled-data systems, we can find the sampled error, $E^*(s) = E(z)$. Adding synchronous samplers at the input and the feedback, we obtain Figure 13.17(b). Pushing $G(s)$ and its input sampler to the right past the pickoff point yields Figure 13.17(c). Using Figure 13.9(a), we can convert each block to its z-transform, resulting in Figure 13.17(d).

From this figure, $E(z) = R(z) - E(z)G(z)$, or

$$E(z) = \frac{R(z)}{1 + G(z)} \tag{13.65}$$

The final value theorem for discrete signals states that

$$e^*(\infty) = \lim_{z \to 1}(1 - z^{-1})E(z) \tag{13.66}$$

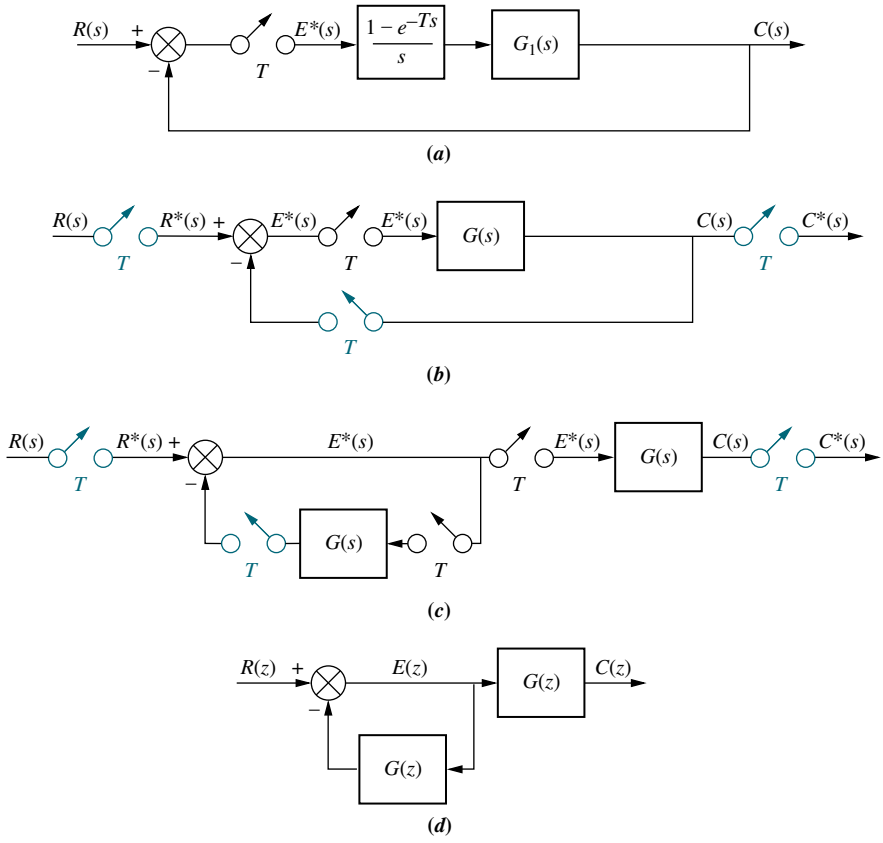


FIGURE 13.17 **a.** Digital feedback control system for evaluation of steady-state errors; **b.** phantom samplers added; **c.** pushing $G(s)$ and its samplers to the right past the pickoff point; **d.** z-transform equivalent system

Note: Phantom samplers are shown in color.

where $e^*(\infty)$ is the final sampled value of $e(t)$, or (alternatively) the final value of $e(kT)$.[4]

Using the final value theorem on Eq. (13.65), we find that the sampled steady-state error, $e^*(\infty)$, for unity negative-feedback systems is

$$e^*(\infty) = \lim_{z\to 1}(1 - z^{-1})E(z) = \lim_{z\to 1}(1 - z^{-1})\frac{R(z)}{1 + G(z)} \tag{13.67}$$

Equation (13.67) must now be evaluated for each input: step, ramp, and parabola.

Unit Step Input

For a unit step input, $R(s) = 1/s$. From Table 13.1,

$$R(z) = \frac{z}{z - 1} \tag{13.68}$$

Substituting Eq. (13.68) into Eq. (13.67), we have

$$e^*(\infty) = \frac{1}{1 + \lim_{z\to 1} G(z)} \tag{13.69}$$

[4] See *Ogata (1987: 59)* for a derivation.

Defining the static error constant, K_p, as

$$K_p = \lim_{z \to 1} G(z) \tag{13.70}$$

we rewrite Eq. (13.69) as

$$e^*(\infty) = \frac{1}{1 + K_p} \tag{13.71}$$

Unit Ramp Input

For a unit ramp input, $R(z) = Tz/(z-1)^2$. Following the procedure for the step input, you can derive the fact that

$$e^*(\infty) = \frac{1}{K_v} \tag{13.72}$$

where

$$K_v = \frac{1}{T} \lim_{z \to 1} (z-1) G(z) \tag{13.73}$$

Unit Parabolic Input

For a unit parabolic input, $R(z) = T^2 z/(z+1)/[2(z-1)^3]$. Similarly,

$$e^*(\infty) = \frac{1}{K_a} \tag{13.74}$$

where

$$K_a = \frac{1}{T^2} \lim_{z \to 1} (z-1)^2 G(z) \tag{13.75}$$

Summary of Steady-State Errors

The equations developed above for $e^*(\infty)$, K_p, K_v, and K_a are similar to the equations developed for analog systems. Whereas multiple pole placement at the origin of the s-plane reduced steady-state errors to zero in the analog case, we can see that multiple pole placement at $z = 1$ reduces the steady-state error to zero for digital systems of the type discussed in this section. This conclusion makes sense when one considers that $s = 0$ maps into $z = 1$ under $z = e^{Ts}$.

For example, for a step input, we see that if $G(z)$ in Eq. (13.69) has one pole at $z = 1$, the limit will become infinite, and the steady-state error will reduce to zero.

For a ramp input, if $G(z)$ in Eq. (13.73) has two poles at $z = 1$, the limit will become infinite, and the error will reduce to zero.

Similar conclusions can be drawn for the parabolic input and Eq. (13.75). Here, $G(z)$ needs three poles at $z = 1$ in order for the steady-state error to be zero. Let us look at an example.

Example 13.9

Finding Steady-State Error

PROBLEM: For step, ramp, and parabolic inputs, find the steady-state error for the feedback control system shown in Figure 13.17(*a*) if

$$G_1(s) = \frac{10}{s(s+1)} \tag{13.76}$$

SOLUTION: First find $G(s)$, the product of the z.o.h. and the plant.

$$G(s) = \frac{10(1-e^{-Ts})}{s^2(s+1)} = 10(1-e^{-Ts})\left[\frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1}\right] \tag{13.77}$$

The z-transform is then

$$\begin{aligned} G(z) &= 10(1-z^{-1})\left[\frac{Tz}{(z-1)^2} - \frac{z}{z-1} + \frac{z}{z-e^{-T}}\right] \\ &= 10\left[\frac{T}{z-1} - 1 + \frac{z-1}{z-e^{-T}}\right] \end{aligned} \tag{13.78}$$

For a step input,

$$K_p = \lim_{z\to 1} G(z) = \infty; \quad e^*(\infty) = \frac{1}{1+K_p} = 0 \tag{13.79}$$

For a ramp input,

$$K_v = \frac{1}{T}\lim_{z\to 1}(z-1)G(z) = 10; \quad e^*(\infty) = \frac{1}{K_v} = 0.1 \tag{13.80}$$

For a parabolic input,

$$K_a = \frac{1}{T^2}\lim_{z\to 1}(z-1)^2 G(z) = 0; \quad e^*(\infty) = \frac{1}{K_a} = \infty \tag{13.81}$$

You will notice that the answers obtained are the same as the results obtained for the analog system. However, since stability depends upon the sampling interval, be sure to check the stability of the system after a sampling interval is established before making steady-state error calculations.

MATLAB
ML

Students who are using MATLAB should now run ch13p6 in Appendix B. You will learn how to use MATLAB to determine K_p, K_v, and K_a in a digital system as well as check the stability. This exercise solves Example 13.9 using MATLAB.

Skill-Assessment Exercise 13.7

WileyPLUS
WPCS
Control Solutions

PROBLEM: For step, ramp, and parabolic inputs, find the steady-state error for the feedback control system shown in Figure 13.17(*a*) if

$$G_1(s) = \frac{20(s+3)}{(s+4)(s+5)}$$

Let $T = 0.1$ second. Repeat for $T = 0.5$ second.

ANSWER: For $T = 0.1$ second, $K_p = 3$, $K_v = 0$, and $K_a = 0$; for $T = 0.5$ second, the system is unstable.

The complete solution is located at www.wiley.com/college/nise.

In this section, we discussed and evaluated the steady-state error of digital systems for step, ramp, and parabolic inputs. The equations for steady-state error parallel those for analog systems. Even the definitions of the static error constants were similar. Poles at the origin of the s-plane for analog systems were replaced with poles at $+1$ on the z-plane to improve the steady-state error. We continue our parallel discussion by moving into a discussion of transient response and the root locus for digital systems.

13.8 Transient Response on the z-Plane

Recall that for analog systems a transient response requirement was specified by selecting a closed-loop, s-plane pole. In Chapter 8, the closed-loop pole was on the existing root locus, and the design consisted of a simple gain adjustment. If the closed-loop pole was not on the existing root locus, then a cascade compensator was designed to reshape the original root locus to go through the desired closed-loop pole. A gain adjustment then completed the design.

In the next two sections, we want to parallel the described analog methods and apply similar techniques to digital systems. For this introductory chapter, we will parallel the discussion through design via gain adjustment. The design of compensation is left to you to pursue in an advanced course.

Chapter 4 established the relationships between transient response and the s-plane. We saw that vertical lines on the s-plane were lines of constant settling time, horizontal lines were lines of constant peak time, and radial lines were lines of constant percent overshoot. In order to draw equivalent conclusions on the z-plane, we now map those lines through $z = e^{sT}$.

The vertical lines on the s-plane are lines of constant settling time and are characterized by the equation $s = \sigma_1 + j\omega$, where the real part, $\sigma_1 = -4/T_s$, is constant and is in the left–half-plane for stability. Substituting this into $z = e^{sT}$, we obtain

$$z = e^{\sigma_1 T} e^{j\omega T} = r_1 e^{j\omega T} \tag{13.82}$$

Equation (13.82) denotes concentric circles of radius r_1. If σ_1 is positive, the circle has a larger radius than the unit circle. On the other hand, if σ_1 is negative, the circle has a smaller radius than the unit circle. The circles of constant settling time, normalized to the sampling interval, are shown in Figure 13.18 with radius $e^{\sigma_1 T} = e^{-4/(T_s/T)}$. Also, $T_s/T = -4/\ln(r)$, where r is the radius of the circle of constant settling time.

The horizontal lines are lines of constant peak time. The lines are characterized by the equation $s = \sigma + j\omega_1$, where the imaginary part, $\omega_1 = \pi/T_p$, is constant. Substituting this into $z = e^{sT}$, we obtain

$$z = e^{\sigma T} e^{j\omega_1 T} = e^{\sigma T} e^{j\theta_1} \tag{13.83}$$

Equation (13.83) represents radial lines at an angle of θ_1. If σ is negative, that section of the radial line lies inside the unit circle. If σ is positive, that section of the radial line lies outside the unit circle. The lines of constant peak time normalized to the sampling interval are shown in Figure 13.18. The angle of each radial line is $\omega_1 T = \theta_1 = \pi/(T_p/T)$, from which $T_p/T = \pi/\theta_1$.

Finally, we map the radial lines of the s-plane onto the z-plane. Remember, these radial lines are lines of constant percent overshoot on the s-plane. From Figure 13.19, these radial lines are represented by

$$\frac{\sigma}{\omega} = -\tan(\sin^{-1}\zeta) = -\frac{\zeta}{\sqrt{1-\zeta^2}} \tag{13.84}$$

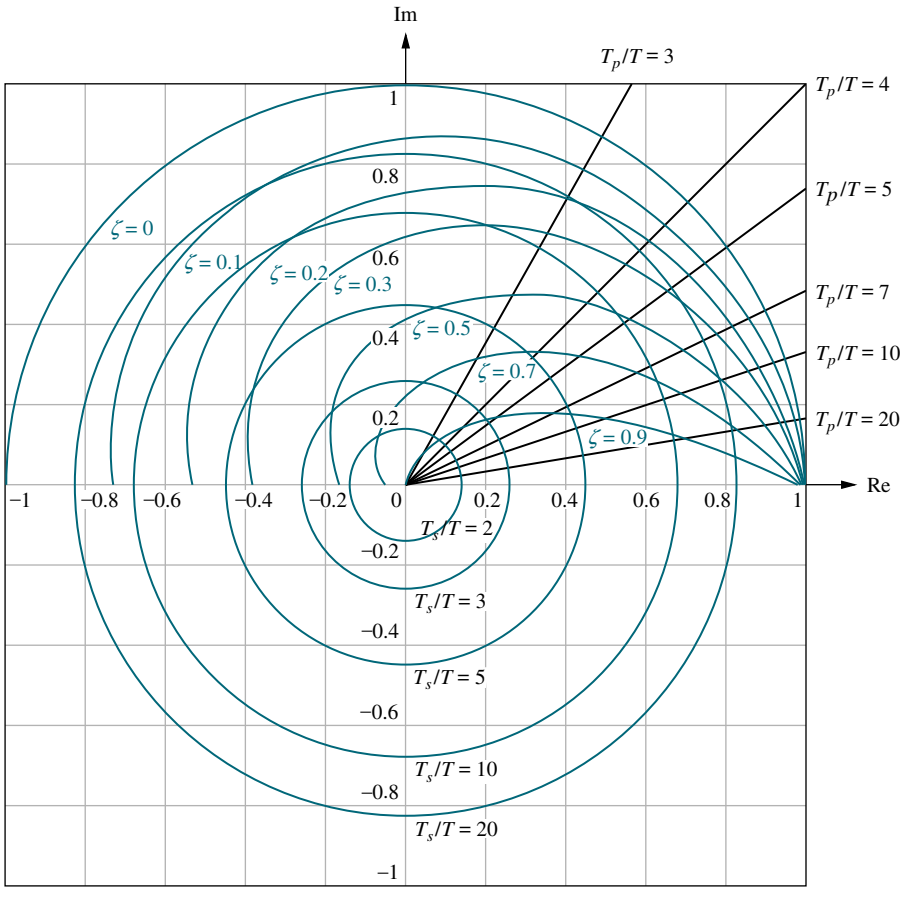


FIGURE 13.18 Constant damping ratio, normalized settling time, and normalized peak time plots on the z-plane

Hence,

$$s = \sigma + j\omega = -\omega \frac{\zeta}{\sqrt{1-\zeta^2}} + j\omega \tag{13.85}$$

Transforming Eq. (13.85) to the z-plane yields

$$z = e^{sT} = e^{-\omega T\left(\zeta/\sqrt{1-\zeta^2}\right)} e^{j\omega T} = e^{-\omega T\left(\zeta/\sqrt{1-\zeta^2}\right)} \angle \omega T \tag{13.86}$$

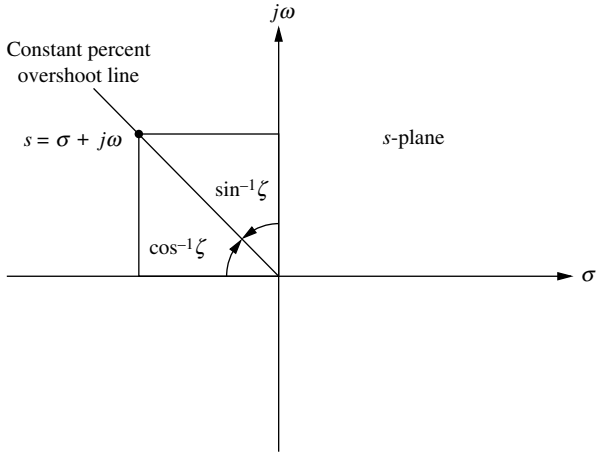


FIGURE 13.19 The s-plane sketch of constant percent overshoot line

Thus, given a desired damping ratio, ζ, Eq. (13.86) can be plotted on the z-plane through a range of ωT as shown in Figure 13.18. These curves can then be used as constant percent overshoot curves on the z-plane.

This section has set the stage for the analysis and design of transient response for digital systems. In the next section, we apply the results to digital systems using the root locus.

13.9 Gain Design on the z-Plane

In this section, we plot root loci and determine the gain required for stability as well as the gain required to meet a transient response requirement. Since the open-loop and closed-loop transfer functions for the generic digital system shown in Figure 13.20 are identical to the continuous system except for a change in variables from s to z, we can use the same rules for plotting a root locus.

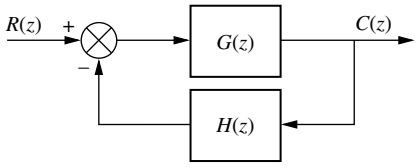


FIGURE 13.20 Generic digital feedback control system

However, from our previous discussion, the region of stability on the z-plane is within the unit circle and not the left half-plane. Thus, in order to determine stability, we must search for the intersection of the root locus with the unit circle rather than the imaginary axis.

In the last section, we derived the curves of constant settling time, peak time, and damping ratio. In order to design a digital system for transient response, we find the intersection of the root locus with the appropriate curves as they appear on the z-plane in Figure 13.18. Let us look at the following example.

Example 13.10

Stability Design via Root Locus

PROBLEM: Sketch the root locus for the system shown in Figure 13.21. Also, determine the range of gain, K, for stability from the root locus plot.

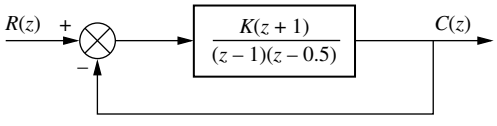


FIGURE 13.21 Digital feedback control for Example 13.10

SOLUTION: Treat the system as if z were s, and sketch the root locus. The result is shown in Figure 13.22. Using the root locus program discussed in Appendix H.2 at www.wiley.com/college/nise, search along the unit circle for 180°. Identification of the gain, K, at this point yields the range of gain for stability. Using the program, we find that the intersection of the root locus with the unit circle is $1\angle 60^\circ$. The gain at this point is 0.5. Hence, the range of gain for stability is $0 < K < 0.5$.

MATLAB
ML

Students who are using MATLAB should now run ch13p7 in Appendix B. You will learn how to use MATLAB to plot a root locus on the z-plane as well as superimpose the unit circle. You will learn how to select interactively the intersection of the root locus and the unit circle to obtain the value of gain for stability. This exercise solves Example 13.10 using MATLAB.

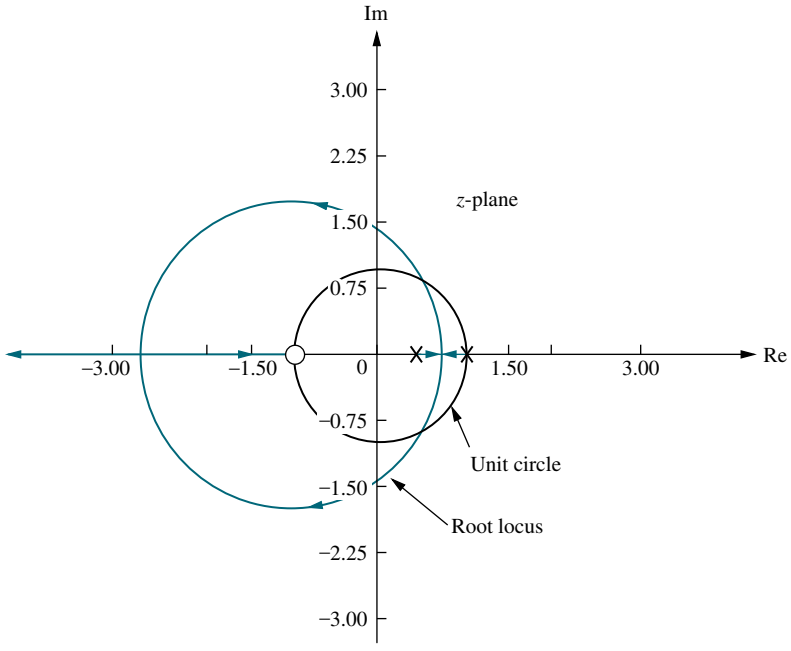


FIGURE 13.22 Root locus for the system of Figure 13.21

In the next example, we design the value of gain, K, in Figure 13.21 to meet a transient response specification. The problem is handled similarly to the analog system design, where we found the gain at the point where the root locus crossed the specified damping ratio, settling time, or peak time curve. In digital systems, these curves are as shown in Figure 13.18. In summary, then, draw the root locus of the digital system and superimpose the curves of Figure 13.18. Then find out where the root locus intersects the desired damping ratio, settling time, or peak time curve and evaluate the gain at that point. In order to simplify the calculations and obtain more accurate results, draw a radial line through the point where the root locus intersects the appropriate curve. Measure the angle of this line and use the root locus program in Appendix H.2 at www.wiley.com/college/nise to search along this radial line for the point of intersection with the root locus.

Example 13.11

Transient Response Design via Gain Adjustment

PROBLEM: For the system of Figure 13.21, find the value of gain, K, to yield a damping ratio of 0.7.

SOLUTION: Figure 13.23 shows the constant damping ratio curves superimposed over the root locus for the system as determined from the last example. Draw a radial line from the origin to the intersection of the root locus with the 0.7 damping ratio curve (a 16.62° line). The root locus program discussed in Appendix H.2 at www.wiley.com/college/nise can now be used to obtain the gain by searching along a 16.62° line for 180°, the intersection with the root locus. The results of the program show that the gain, K, is 0.0627 at $0.719 + j\,0.215$, the point where the 0.7 damping ratio curve intersects the root locus.

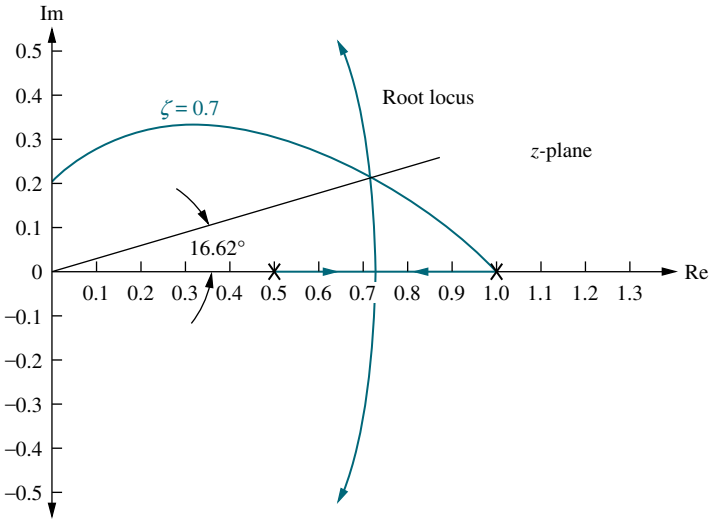


FIGURE 13.23 Root locus for the system of Figure 13.21 with constant 0.7 damping ratio curve

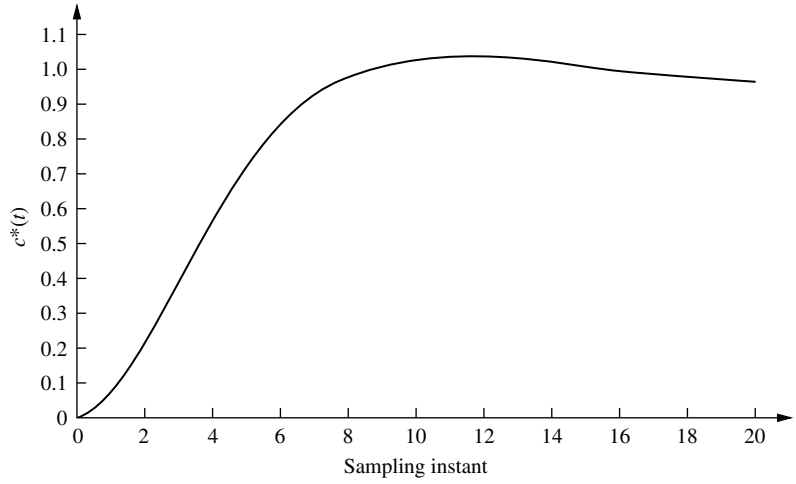
We can now check our design by finding the unit sampled step response of the system of Figure 13.21. Using our design, $K = 0.0627$, along with $R(z) = z/(z-1)$, a sampled step input, we find the sampled output to be

$$C(z) = \frac{R(z)G(z)}{1+G(z)} = \frac{0.0627z^2 + 0.0627z}{z^3 - 2.4373z^2 + 2z - 0.5627} \tag{13.87}$$

Performing the indicated division, we obtain the output valid at the sampling instants, as shown in Figure 13.24. Since the overshoot is approximately 5%, the requirement of a 0.7 damping ratio has been met. You should remember, however, that the plot is valid only at integer values of the sampling instants.

MATLAB
ML

Students who are using MATLAB should now run ch13p8 in Appendix B. You will learn how to use MATLAB to plot a root locus on the z-plane as well as superimpose a grid of damping ratio curves. You will learn how to obtain the gain and a closed-loop step response of a digital system after interactively selecting the operating point on the root locus. This exercise solves Example 13.11 using MATLAB.



Note: Valid only at integer values of sampling instant

FIGURE 13.24 Sampled step response of the system of Figure 13.21 with $K = 0.0627$

Skill-Assessment Exercise 13.8

TryIt 13.3

Use MATLAB, the Control System Toolbox, and the following statements to solve Skill-Assessment Exercise 13.8.

```
Gz=zpk(-0.5,[0.25,0.75],...
 1,[])
rlocus(Gz)
zgrid(0.5,[])
[K,p]=rlocfind (Gz)
```

Note: When the root locus appears, click on the intersection of the 0.5 damping ratio curve and the root locus to calculate the gain.

PROBLEM: For the system of Figure 13.20 where $H(z) = 1$ and

$$G(z) = \frac{K(z + 0.5)}{(z - 0.25)(z - 0.75)}$$

find the value of gain, K, to yield a damping ratio of 0.5.

ANSWER: $K = 0.31$

The complete solution is at www.wiley.com/college/nise.

Simulink SL

MATLAB's Simulink provides an alternative method of simulating digital systems to obtain the time response. Students who are performing the MATLAB exercises and want to explore the added capability of Simulink should now consult Appendix C, MATLAB's Simulink Tutorial. Example C.4 in the tutorial shows how to use Simulink to simulate digital systems.

Gui Tool GUIT

MATLAB's LTI Viewer provides another method of simulating digital systems to obtain the time response. Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's LTI Viewer should now consult Appendix E at www.wiley.com/college/nise, which contains a tutorial on the LTI Viewer as well as some examples. One of the illustrative examples, Example E.5, finds the closed-loop step response of a digital system using the LTI Viewer.

In this section, we used the root locus and gain adjustment to design the transient response of a digital system. This method suffers the same drawbacks as when it was applied to analog systems; namely, if the root locus does not intersect a desired design point, then a simple gain adjustment will not accomplish the design objective. Techniques to design compensation for digital systems can then be applied.

13.10 Cascade Compensation via the *s*-Plane

In previous sections of this chapter, we analyzed and designed digital systems directly in the z-domain up to and including design via gain adjustment. We are now ready to design digital compensators, such as those covered in Chapters 9 and 11. Rather than continuing on this path of design directly in the z-domain, we depart by covering analysis and design techniques that allow us to make use of previous chapters by designing on the s-plane and then transforming our s-plane design to a

digital implementation. We covered one aspect of *s*-plane analysis in Section 13.6, where we used a bilinear transformation to analyze stability. We now continue with *s*-plane analysis and design by applying it to cascade compensator design. Direct design of compensators on the *z*-plane is left for a dedicated course in digital control systems.

Cascade Compensation

In order to perform design in the *s*-plane and then convert the continuous compensator to a digital compensator, we need a bilinear transformation that will preserve, at the sampling instants, the response of the continuous compensator. The bilinear transformation covered in Section 13.6 will not meet that requirement. A bilinear transformation that can be performed with hand calculations and yields a digital transfer function whose output response at the sampling instants is approximately the same as the equivalent analog transfer function is called the *Tustin transformation*. This transformation is used to transform the continuous compensator, $G_c(s)$, to the digital compensator, $G_c(z)$. The Tustin transformation is given by[5]

$$s = \frac{2(z-1)}{T(z+1)} \tag{13.88}$$

and its inverse by

$$z = \frac{-\left(s + \frac{2}{T}\right)}{\left(s - \frac{2}{T}\right)} = \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s} \tag{13.89}$$

As the sampling interval, T, gets smaller (higher sampling rate), the designed digital compensator's output yields a closer match to the analog compensator. If the sampling rate is not high enough, there is a discrepancy at higher frequencies between the digital and analog filters' frequency responses. Methods are available to correct the discrepancy, but they are beyond the scope of our discussion. The interested reader should investigate the topic of *prewarping*, covered in books dedicated to digital control and listed in the Bibliography at the end of this chapter.

Astrom and Wittenmark (*1984*) have developed a guideline for selecting the sampling interval, *T*. Their conclusion is that the value of T in seconds should be in the range $0.15/\omega_{\Phi_M}$ to $0.5/\omega_{\Phi_M}$, where ω_{Φ_M} is the zero dB frequency (rad/s) of the magnitude frequency response curve for the cascaded analog compensator and plant.

In the following example, we will design a compensator, $G_c(s)$, to meet the required performance specifications. We will then use the Tustin transformation to obtain the model for an equivalent digital controller. In the next section, we will show how to implement the digital controller.

[5] See *Ogata (1987: 315–318)* for a derivation.

Example 13.12

Digital Cascade Compensator Design

PROBLEM: For the digital control system of Figure 13.25(a), where

$$G_p(s) = \frac{1}{s(s+6)(s+10)} \tag{13.90}$$

design a digital lead compensator, $G_c(z)$, as shown in Figure 13.25(c), so that the system will operate with 20% overshoot and a settling time of 1.1 seconds. Create your design in the s-domain and transform the compensator to the z-domain.

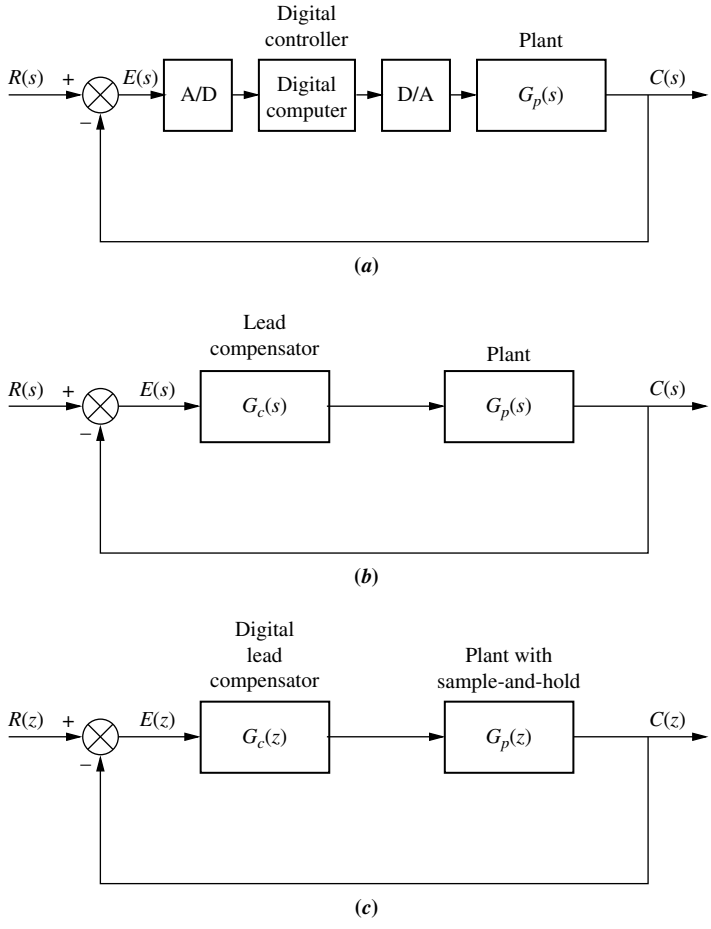


FIGURE 13.25 **a.** Digital control system showing the digital computer performing compensation; **b.** continuous system used for design; **c.** transformed digital system

SOLUTION: Using Figure 13.25(b), design a lead compensator using the techniques described in Chapter 9 or 11. The design was created as part of Example 9.6, where we found that the lead compensator was

$$G_c(s) = \frac{1977(s+6)}{(s+29.1)} \tag{13.91}$$

Using Eqs. (13.90) and (13.91), we find that the zero dB frequency, ω_{Φ_M}, for $G_p(s)G_c(s)$ is 5.8 rad/s. Using the guideline described by *Astrom and Wittenmark (1984)*, the lowest value of T should be in the range $0.15/\omega_{\Phi_M} = 0.026$ to $0.5/\omega_{\Phi_M} = 0.086$ second. Let us use $T = 0.01$ second.

Substituting Eq. (13.88) into Eq. (13.91) with $T = 0.01$ second yields

$$G_c(z) = \frac{1778z - 1674}{z - 0.746} \tag{13.92}$$

The z-transform of the plant and zero-order hold, found by the method discussed in Section 13.4 with $T = 0.01$ second, is

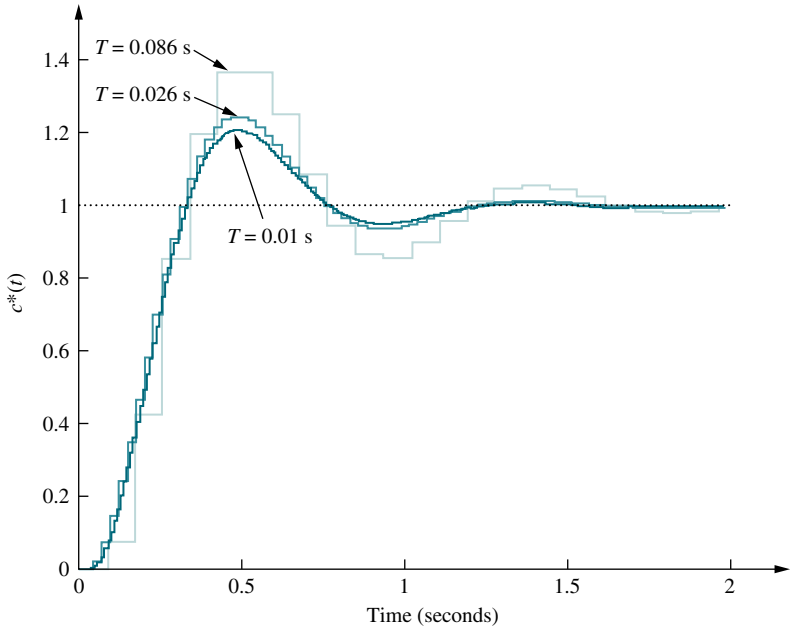
$$G_p(z) = \frac{(1.602 \times 10^{-7}z^2) + (6.156 \times 10^{-7}z) + (1.478 \times 10^{-7})}{z^3 - 2.847z^2 + 2.699z - 0.8521} \tag{13.93}$$

The time response in Figure 13.26 ($T = 0.01$ s) shows that the compensated closed-loop system meets the transient response requirements. The figure also shows the response for a compensator designed with sampling times at the extremes of Astrom and Wittenmark's guideline.

MATLAB

ML

`Students who are using MATLAB should now run ch13p9 in Appendix B. You will learn how to use MATLAB to design a digital lead compensator using the Tustin transformation. This exercise solves Example 13.12 using MATLAB.`



Note: Valid only at integer values of sampling instant

FIGURE 13.26 Closed-loop response for the compensated system of Example 13.12 showing effect of three different sampling frequencies

Skill-Assessment Exercise 13.9

WileyPLUS
WPCS
Control Solutions

PROBLEM: In Example 11.3, a lead compensator was designed for a unity feedback system whose plant was

$$G(s) = \frac{100K}{s(s+36)(s+100)}$$

The design specifications were as follows: percent overshoot = 20%, peak time 0.1 second, and $K_v = 40$. In order to meet the requirements, the design yielded $K = 1440$ and a lead compensator,

$$G_c(s) = 2.38\,\frac{s+25.3}{s+60.2}$$

If the system is to be computer controlled, find the digital controller, $G_c(z)$.

ANSWER: $G_c(z) = 2.34\,\dfrac{z-0.975}{z-0.9416}$, $T = 0.001$ second

The complete solution is at www.wiley.com/college/nise.

Now that we have learned how to design a digital cascade compensator, $G_c(z)$, the next section will teach us how to use the digital computer to implement it.

13.11 Implementing the Digital Compensator

The controller, $G_c(z)$, can be implemented directly via calculations within the digital computer in the forward path as shown in Figure 13.27. Let us now derive a numerical algorithm that the computer can use to emulate the compensator. We will find an expression for the computer's sampled output, $x^*(t)$, whose transforms are shown in Figure 13.27 as $X(z)$. We will see that this expression can be used to program the digital computer to emulate the compensator.

Consider a second-order compensator, $G_c(z)$,

$$G_c(z) = \frac{X(z)}{E(z)} = \frac{a_3z^3 + a_2z^2 + a_1z + a_0}{b_2z^2 + b_1z + b_0} \tag{13.94}$$

Cross-multiplying,

$$(b_2z^2 + b_1z + b_0)X(z) = (a_3z^3 + a_2z^2 + a_1z + a_0)E(z) \tag{13.95}$$

Solving for the term with the highest power of z operating on the output, $X(z)$,

$$b_2z^2X(z) = (a_3z^3 + a_2z^2 + a_1z + a_0)E(z) - (b_1z + b_0)X(z) \tag{13.96}$$

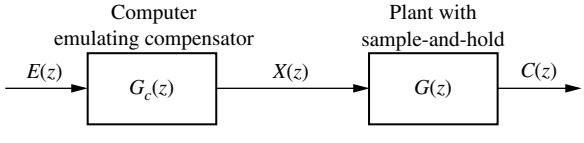


FIGURE 13.27 Block diagram showing computer emulation of a digital compensator

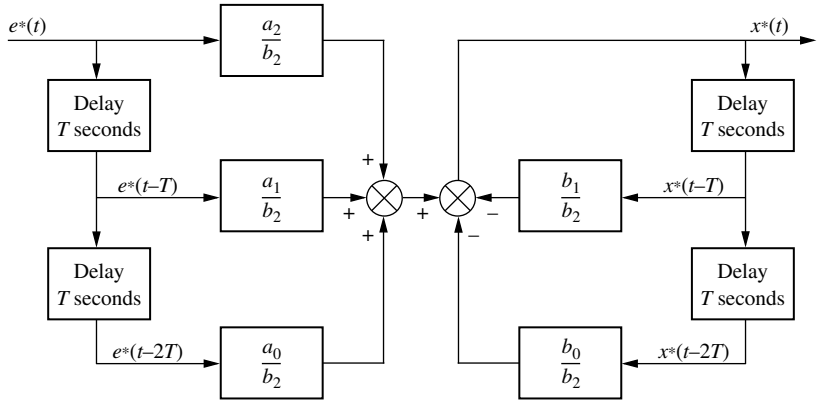


FIGURE 13.28 Flowchart for a second-order digital compensator (Reprinted with permission of John Wiley & Sons, Inc.)

Dividing by the coefficient of $X(z)$ on the left-hand side of Eq. (13.96) yields

$$X(z) = \left(\frac{a_3}{b_2}z + \frac{a_2}{b_2} + \frac{a_1}{b_2}z^{-1} + \frac{a_0}{b_2}z^{-2}\right)E(z) - \left(\frac{b_1}{b_2}z^{-1} + \frac{b_0}{b_2}z^{-2}\right)X(z) \tag{13.97}$$

Finally, taking the inverse z-transform,

$$\begin{aligned} x^*(t) \quad &= \frac{a_3}{b_2}e^*(t+T) + \frac{a_2}{b_2}e^*(t) + \frac{a_1}{b_2}e^*(t-T) + \frac{a_0}{b_2}e^*(t-2T) \\ &\quad - \frac{b_1}{b_2}x^*(t-T) - \frac{b_0}{b_2}x^*(t-2T) \end{aligned} \tag{13.98}$$

We can see from this equation that the present sample of the compensator output, $x^*(t)$, is a function of future $(e^*(t+T))$ present $(e^*(t))$ and past $(e^*(t-T))$ and $e^*(t-2T))$ samples of $e(t)$, along with past values of the output, $x^*(t-T)$ and $x^*(t-2T)$. Obviously, if we are to physically realize this compensator, the output sample cannot be dependent upon future values of the input. Hence, to be physically realizable, a_3 must equal zero for the future value of $e(t)$ to be zero. We conclude that the numerator of the compensator's transfer function must be of equal or lower order than the denominator in order that the compensator be physically realizable.

Now assume that a_3 does indeed equal zero. Equation (13.98) now becomes

$$x^*(t) = \frac{a_2}{b_2}e^*(t) + \frac{a_1}{b_2}e^*(t-T) + \frac{a_0}{b_2}e^*(t-2T) - \frac{b_1}{b_2}x^*(t-T) - \frac{b_0}{b_2}x^*(t-2T) \tag{13.99}$$

Hence, the output sample is a function of current and past input samples of the input as well as past samples of the output. Figure 13.28 shows the flowchart of the compensator from which a program can be written for the digital computer.[6] The figure shows that the compensator can be implemented by storing several successive values of the input and output. The output is then formed by a weighted linear combination of these stored variables. Let us now look at a numerical example.

[6] For an excellent discussion on basic flowcharts to represent digital compensators, including the representation shown in Figure 13.28 and alternative flowcharts with half as many delays, see *Chassaing (1999, pp. 135–143)*.

Example 13.13

Digital Cascade Compensator Implementation

PROBLEM: Develop a flowchart for the digital compensator defined by Eq. (13.100).

$$G_c(z) = \frac{X(z)}{E(z)} = \frac{z + 0.5}{z^2 - 0.5z + 0.7} \tag{13.100}$$

SOLUTION: Cross-multiply and obtain

$$(z^2 - 0.5z + 0.7)X(z) = (z + 0.5)E(z) \tag{13.101}$$

Solve for the highest power of z operating on the output, $X(z)$,

$$z^2X(z) = (z + 0.5)E(z) - (-0.5z + 0.7)X(z) \tag{13.102}$$

Solving for $X(z)$ on the left-hand side,

$$X(z) = (z^{-1} + 0.5z^{-2})E(z) - (-0.5z^{-1} + 0.7z^{-2})X(z) \tag{13.103}$$

Implementing Eq. (13.103) with the flowchart of Figure 13.29 completes the design.

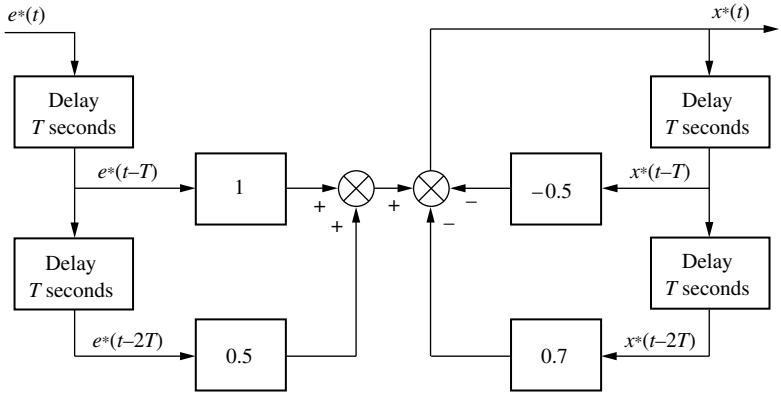


FIGURE 13.29 Flowchart to implement $G_c(z) = \dfrac{z + 0.5}{z^2 - 0.5z + 0.7}$ (Reprinted with permission of John Wiley & Sons, Inc.)

Skill-Assessment Exercise 13.10

PROBLEM: Draw a flowchart from which the compensator

$$G_c(z) = \frac{1899z^2 - 3761z + 1861}{z^2 - 1.908z + 0.9075}$$

can be programmed if the sampling interval is 0.1 second.

ANSWER: The complete solution is at www.wiley.com/college/nise.

In this section, we learned how to implement a digital compensator. The resulting flowchart can serve as the design of a digital computer program for the computer in the loop. The design consists of delays that can be thought of as storage for each sampled value of input and output. The stored values are weighted and added. The engineer then can implement the design with a computer program.

In the next section, we will put together the concepts of this chapter as we apply the principles of digital control system design to our antenna azimuth control system.

Case Studies

Antenna Control: Transient Design via Gain

Design
D

We now demonstrate the objectives of this chapter by turning to our ongoing antenna azimuth position control system. We will show where the computer is inserted in the loop, model the system, and design the gain to meet a transient response requirement. Later, we will design a digital cascade compensator.

The computer will perform two functions in the loop. First, the computer will be used as the input device. It will receive digital signals from the keyboard in the form of commands, and digital signals from the output for closed-loop control. The keyboard will replace the input potentiometer, and an analog-to-digital (A/D) converter along with a unity gain feedback transducer will replace the output potentiometer.

Figure 13.30(a) shows the original analog system, and Figure 13.30(b) shows the system with the computer in the loop. Here the computer is receiving digital signals from two sources: (1) the input via the keyboard or other tracking commands and (2) the output via an A/D converter. The plant is receiving signals from the digital computer via a digital-to-analog (D/A) converter and the sample-and-hold.

Figure 13.30(b) shows some simplifying assumptions we have made. The power amplifier's pole is assumed to be far enough away from the motor's pole that we can represent the power amplifier as a pure gain equal to its dc gain of unity. Also, we have absorbed any preamplifier and potentiometer gain in the computer and its associated D/A converter.

PROBLEM: Design the gain for the antenna azimuth position control system shown in Figure 13.30(b) to yield a closed-loop damping ratio of 0.5. Assume a sampling interval of $T = 0.1$ second.

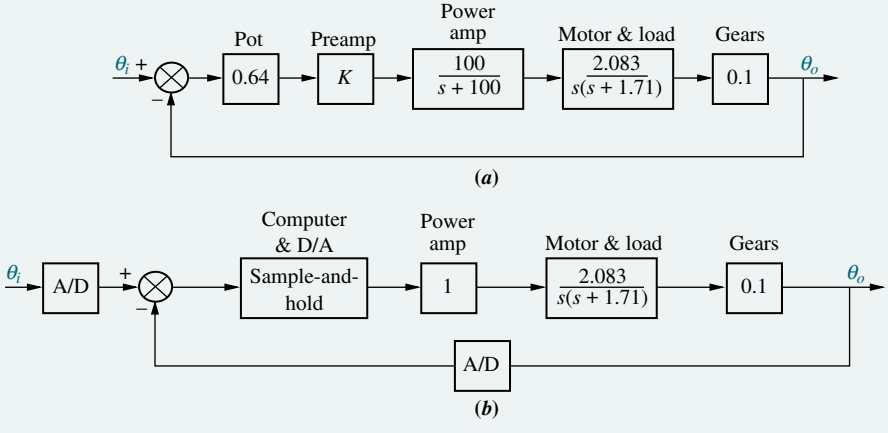


FIGURE 13.30 Antenna control system: **a.** analog implementation; **b.** digital implementation

SOLUTION: **Modeling the System:** Our first objective is to model the system in the z-domain. The forward transfer function, $G(s)$, which includes the sample-and-hold, power amplifier, motor and load, and the gears, is

$$G(s) = \frac{1 - e^{-Ts}}{s} \frac{0.2083}{s(s + a)} = \frac{0.2083}{a}(1 - e^{Ts}) \frac{a}{s^2(s + a)} \tag{13.104}$$

where $a = 1.71$, and $T = 0.1$.

Since the z-transform of $(1 - e^{-Ts})$ is $(1 - z^{-1})$ and, from Example 13.6, the z-transform of $a/[s^2(s + a)]$ is

$$z\left\{\frac{a}{s^2(s + a)}\right\} = \left[\frac{Tz}{(z - 1)^2} - \frac{(1 - e^{-aT})z}{a(z - 1)(z - e^{-aT})}\right] \tag{13.105}$$

the z-transform of the plant, $G(z)$, is

$$\begin{aligned} G(z) &= \frac{0.2083}{a}(1 - z^{-1})z\left\{\frac{a}{s^2(s + a)}\right\} \\ &= \frac{0.2083}{a^2}\left[\frac{[aT - (1 - e^{-aT})]z + [(1 - e^{-aT}) - aTe^{-aT}]}{(z - 1)(z - e^{-aT})}\right] \end{aligned} \tag{13.106}$$

Substituting the values for a and T, we obtain

$$G(z) = \frac{9.846 \times 10^{-4}(z + 0.945)}{(z - 1)(z - 0.843)} \tag{13.107}$$

Figure 13.31 shows the computer and plant as part of the digital feedback control system.

Designing for Transient Response: Now that the modeling in the z-domain is complete, we can begin to design the system for the required transient response. We superimpose the root locus over the constant damping ratio curves in the z-plane, as shown in Figure 13.32. A line drawn from the origin to the intersection forms an 8.58° angle. Searching along this line for 180°, we find the intersection to be $(0.915 + j0.138)$, with a loop gain, $9.846 \times 10^{-4}K$, of 0.0135. Hence, $K = 13.71$.

Checking the design by finding the unit sampled step response of the closed-loop system yields the plot of Figure 13.33, which exhibits 20% overshoot ($\zeta = 0.456$).

CHALLENGE: We now give you a case study to test your knowledge of this chapter's objectives: You are given the antenna azimuth position control system shown on the front endpapers, Configuration 2. Do the following:

a. Convert the system into a digital system with $T = 0.1$ second. For the purposes of the conversion, assume that the potentiometers are replaced with unity gain transducers. Neglect power amplifier dynamics.

b. Design the gain, K, for 16.3% overshoot.

c. For your designed value of gain, find the steady-state error for a unit ramp input.

MATLAB ML

d. `Repeat Part` **b** `using MATLAB.`

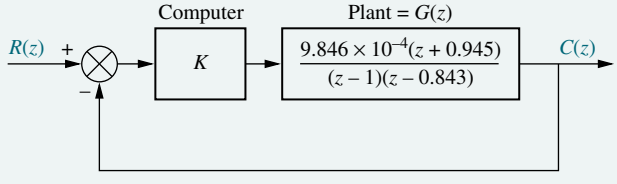


FIGURE 13.31 Analog antenna azimuth position control system converted to a digital system

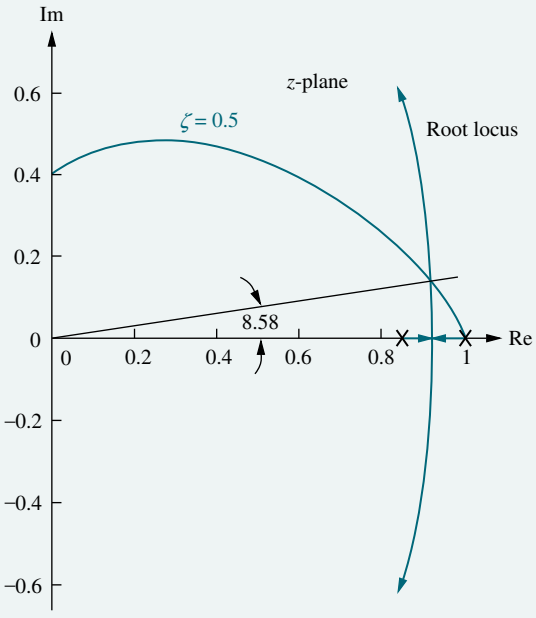
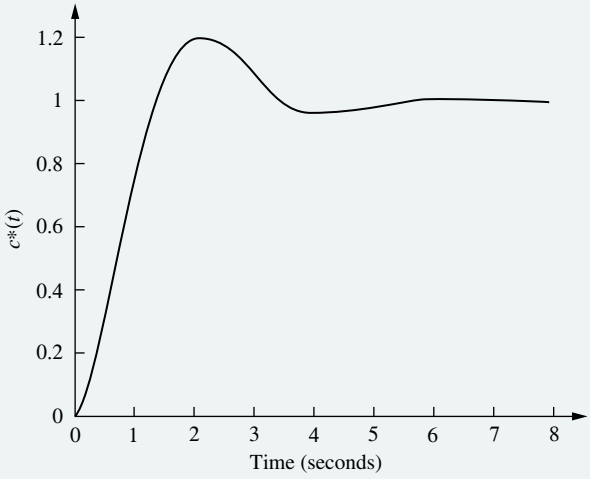


FIGURE 13.32 Root locus superimposed over constant damping ratio curve



Note: Valid only at integer values of sampling instant

FIGURE 13.33 Sampled step response of the antenna azimuth position control system

Antenna Control: Digital Cascade Compensator Design

Design
D

PROBLEM: Design a digital lead compensator to reduce the settling time by a factor of 2.5 from that obtained for the antenna azimuth control system in the previous Case Study problem in this chapter.

SOLUTION: Figure 13.34 shows a simplified block diagram of the continuous system, neglecting power amplifier dynamics and assuming that the potentiometers are replaced with unity gain transducers as previously explained.

We begin with an s-plane design. From Figure 13.33, the settling time is about 5 seconds. Thus, our design requirements are a settling time of 2 seconds and a damping ratio of 0.5. The natural frequency is $\omega_n = 4/(\zeta T_s) = 4$ rad/s. The compensated dominant poles are located at $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -2 \pm j3.464$.

Designing a lead compensator zero to cancel the plant pole on the s-plane at -1.71 yields a lead compensator pole at -4. Hence the lead compensator is given by

$$G_c(s) = \frac{s + 1.71}{s + 4} \tag{13.108}$$

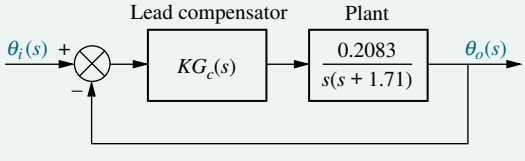


FIGURE 13.34 Simplified block diagram of antenna azimuth control system

Using root locus to evaluate the gain, K, at the design point yields $0.2083K = 16$, or $K = 76.81$.

We now select an appropriate sampling frequency as described in Section 13.10. Using the cascaded compensator,

$$KG_c(s) = \frac{76.81(s + 1.71)}{(s + 4)} \tag{13.109}$$

and plant,

$$G_p(s) = \frac{0.2083}{s(s + 1.71)} \tag{13.110}$$

the equivalent forward-path transfer function, $G_e(s) = KG_c(s)G_p(s)$, is

$$G_e(s) = \frac{16}{s(s + 4)} \tag{13.111}$$

The magnitude frequency response of Eq. (13.111) is 0 dB at 3.1 rad/s. Thus, from Section 13.10, the value of the sampling interval, T, should be in the range $0.15/\omega_{\Phi_M} = 0.05$ to $0.5/\omega_{\Phi_M} = 0.16$ second. Let us choose a smaller value, say $T = 0.025$ second.

Substituting Eq. (13.88) into Eq. (13.111), where $T = 0.025$, yields the digital compensator

$$KG_c(z) = \frac{74.72z - 71.59}{z - 0.9048} \tag{13.112}$$

In order to simulate the digital system, we calculate the z-transform of the plant in Figure 13.34 in cascade with a zero-order sample-and-hold. The z-transform of the sampled plant is evaluated by the method discussed in Section 13.4 using $T = 0.025$. The result is

$$G_p(z) = \frac{6.418 \times 10^{-5}z + 6.327 \times 10^{-5}}{z^2 - 1.958z + 0.9582} \tag{13.113}$$

The step response in Figure 13.35 shows approximately 20% overshoot and a settling time of 2.1 seconds for the closed-loop digital system.

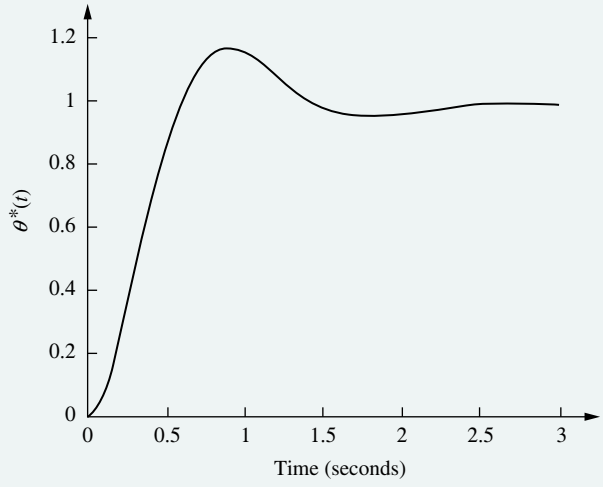


FIGURE 13.35 Closed-loop digital step response for antenna control system with a lead compensator

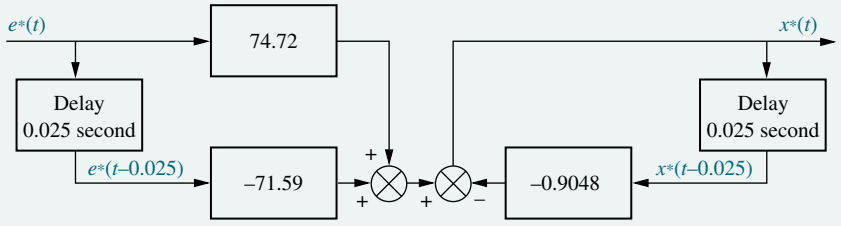


FIGURE 13.36 Flowchart for digital lead compensator. (Reprinted with permission of John Wiley & Sons, Inc.)

We conclude the design by obtaining a flowchart for the digital compensator. Using Eq. (13.112), where we define $KG_c(z) = X(z)/E(z)$, and cross-multiplying yields

$$(z - 0.9048)X(z) = (74.72z - 71.59)E(z) \tag{13.114}$$

Solving for the highest power of z operating on $X(z)$,

$$zX(z) = (74.72z - 71.59)E(z) + 0.9048X(z) \tag{13.115}$$

Solving for $X(z)$,

$$X(z) = (74.72 - 71.59z^{-1})E(z) + 0.9048z^{-1}X(z) \tag{13.116}$$

Implementing Eq. (13.116) as a flowchart yields Figure 13.36.

CHALLENGE: You are now given a case study to test your knowledge of this chapter's objectives. You are given the antenna azimuth position control system shown on the front endpapers, Configuration 2. Replace the potentiometers with unity gain transducers, neglect power amplifier dynamics, and do the following:

a. Design a digital lead compensator to yield 10% overshoot with a 1-second peak time. Design in the s-plane and use the Tustin transformation to specify and implement a digital compensator. Choose an appropriate sampling interval.

b. Draw a flowchart for your digital lead compensator.

c. `Repeat Part` **a** `using MATLAB.`

MATLAB ML

Summary

In this chapter, we covered the design of digital systems using classical methods. State-space techniques were not covered. However, you are encouraged to pursue this topic in a course dedicated to sampled-data control systems.

We looked at the advantages of digital control systems. These systems can control numerous loops at reduced cost. System modifications can be implemented with software changes rather than hardware changes.

Typically, the digital computer is placed in the forward path preceding the plant. Digital-to-analog and analog-to-digital conversion is required within the system to ensure compatibility of the analog and digital signals throughout the system. The digital computer in the loop is modeled as a sample-and-hold network along with any compensation that it performs.

Throughout the chapter, we saw direct parallels to the methods used for s-plane analysis of transients, steady-state errors, and the stability of analog systems.

The parallel is made possible by the z-transform, which replaces the Laplace transform as the transform of choice for analyzing sampled-data systems. The z-transform allows us to represent sampled waveforms at the sampling instants. We can handle sampled systems as easily as continuous systems, including block diagram reduction, since both signals and systems can be represented in the z-domain and manipulated algebraically. Complex systems can be reduced to a single block through techniques that parallel those used with the s-plane. Time responses can be obtained through division of the numerator by the denominator without the partial-fraction expansion required in the s-domain.

Digital systems analysis parallels the s-plane techniques in the area of stability. The unit circle becomes the boundary of stability, replacing the imaginary axis.

We also found that the concepts of root locus and transient response are easily carried into the z-plane. The rules for sketching the root locus do not change. We can map points on the s-plane into points on the z-plane and attach transient response characteristics to the points. Evaluating a sampled-data system shows that the sampling rate, in addition to gain and load, determines the transient response.

Cascade compensators also can be designed for digital systems. One method is to first design the compensator on the s-plane or via frequency response techniques described in Chapters 9 and 11, respectively. Then the resulting design is transformed to a digital compensator using the Tustin transformation. Designing cascade compensation directly on the z-plane is an alternative method that can be used. However, these techniques are beyond the scope of this book.

This introductory control systems course is now complete. You have learned how to analyze and design linear control systems using frequency-domain and state-space techniques. This course is only a beginning. You may consider furthering your study of control systems by taking advanced courses in digital, nonlinear, and optimal control, where you will learn new techniques for analyzing and designing classes of systems not covered in this book. We hope we have whetted your appetite to continue your education in control systems engineering.

Review Questions

1. Name two functions that the digital computer can perform when used with feedback control systems.
2. Name three advantages of using digital computers in the loop.
3. Name two important considerations in analog-to-digital conversion that yield errors.
4. Of what does the block diagram model for a computer consist?
5. What is the z-transform?
6. What does the inverse z-transform of a time waveform actually yield?
7. Name two methods of finding the inverse z-transform.
8. What method for finding the inverse z-transform yields a closed-form expression for the time function?
9. What method for finding the inverse z-transform immediately yields the values of the time waveform at the sampling instants?
10. In order to find the z-transform of a $G(s)$, what must be true of the input and the output?
11. If input $R(z)$ to system $G(z)$ yields output $C(z)$, what is the nature of $c(t)$?

12. If a time waveform, $c(t)$, at the output of system $G(z)$ is plotted using the inverse z-transform, and a typical second-order response with damping ratio = 0.5 results, can we say that the system is stable?
13. What must exist in order for cascaded sampled-data systems to be represented by the product of their pulse transfer functions, $G(z)$?
14. Where is the region for stability on the z-plane?
15. What methods for finding the stability of digital systems can replace the Routh-Hurwitz criterion for analog systems?
16. To drive steady-state errors in analog systems to zero, a pole can be placed at the origin of the s-plane. Where on the z-plane should a pole be placed to drive the steady-state error of a sampled system to zero?
17. How do the rules for sketching the root locus on the z-plane differ from those for sketching the root locus on the s-plane?
18. Given a point on the z-plane, how can one determine the associated percent overshoot, settling time, and peak time?
19. Given a desired percent overshoot and settling time, how can one tell which point on the z-plane is the design point?
20. Describe how digital compensators can be designed on the s-plane.
21. What characteristic is common between a cascade compensator designed on the s-plane and the digital compensator to which it is converted?

Problems

1. Derive the z-transforms for the time functions listed below. Do not use any z-transform tables. Use the plan $f(t) \rightarrow f^*(t) \rightarrow F^*(s) \rightarrow F(z)$, followed by converting $F(z)$ into closed form making use of the fact that $1/(1-z^{-1}) = 1 + z^{-1} + z^{-2} + z^{-3} + \cdots$. Assume ideal sampling. [Section: 13.3]
   a. $e^{-at}u(t)$
   b. $u(t)$
   c. $t^2e^{-at}u(t)$
   d. $\cos \omega t\, u(t)$

2. Repeat all parts of Problem 1 using MATLAB and MATLAB's Symbolic Math Toolbox.

Symbolic Math SM

3. For each $F(z)$, find $f(kT)$ using partial-fraction expansion. [Section: 13.3]
   a. $F(z) = \dfrac{z(z+3)(z+5)}{(z-0.4)(z-0.6)(z-0.8)}$
   b. $F(z) = \dfrac{(z+0.2)(z+0.4)}{(z-0.1)(z-0.5)(z-0.9)}$
   c. $F(z) = \dfrac{(z+1)(z+0.3)(z+0.4)}{z(z-0.2)(z-0.5)(z-0.7)}$

4. Repeat all parts of Problem 3 using MATLAB and MATLAB's Symbolic Math Toolbox.

Symbolic Math SM

5. For each $F(z)$ in Problem 3, do the following: [Section: 13.3]
   a. Find $f(kT)$ using the power series expansion.
   b. Check your results against your answers from Problem 3.

6. Using partial-fraction expansion and Table 13.1, find the z-transform for each $G(s)$ shown below if $T = 0.5$ second. [Section: 13.3]
   a. $G(s) = \dfrac{(s+4)}{(s+2)(s+5)}$
   b. $G(s) = \dfrac{(s+1)(s+2)}{s(s+3)(s+4)}$
   c. $G(s) = \dfrac{20}{(s+3)(s^2+6s+25)}$
   d. $G(s) = \dfrac{15}{s(s+1)(s^2+10s+81)}$

WileyPLUS WPCS Control Solutions

Symbolic Math SM

7. Repeat all parts of Problem 6 using MATLAB and MATLAB's Symbolic Math Toolbox.

8. Find $G(z) = C(z)/R(z)$ for each of the block diagrams shown in Figure P13.1 if $T = 0.3$ second. [Section: 13.4]

9. Find $T(z) = C(z)/R(z)$ for each of the systems shown in Figure P13.2. [Section: 13.5]

10. Find $C(z)$ in general terms for the digital system shown in Figure P13.3. [Section: 13.5]

11. Find the closed-loop transfer function, $T(z) = C(z)/R(z)$, for the system shown in Figure P13.4. [Section: 13.5]

WileyPLUS WPCS Control Solutions

12. Given the system in Figure P13.5, find the range of sampling interval, T, that will keep the system stable. [Section: 13.6]

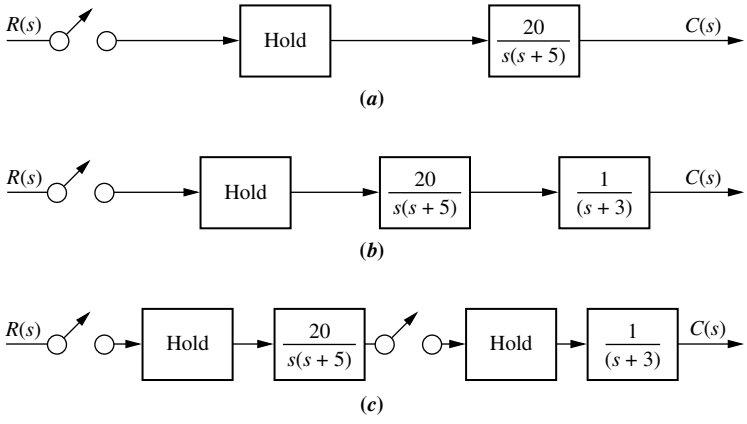


FIGURE P13.1

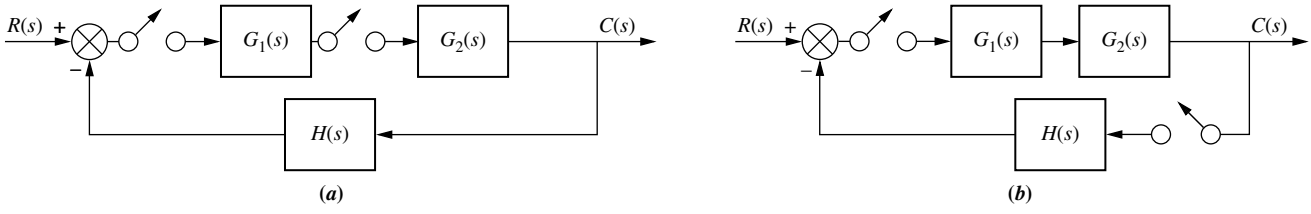


FIGURE P13.2

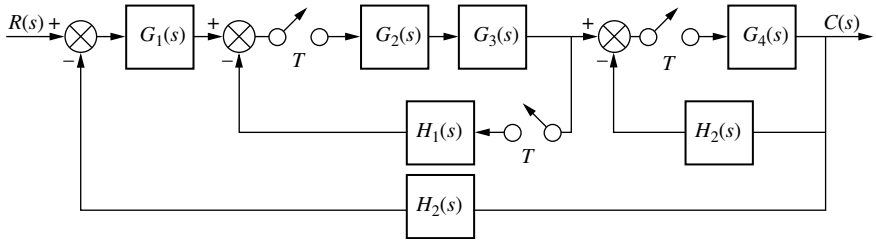


FIGURE P13.3

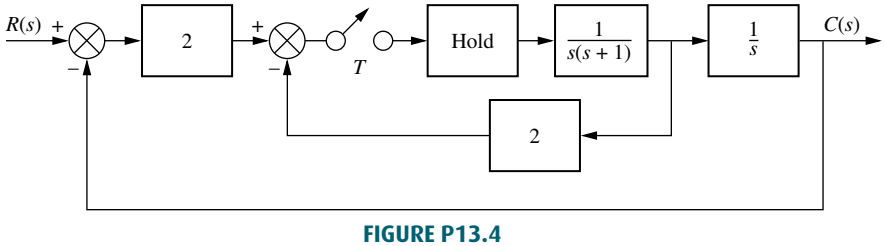


FIGURE P13.4

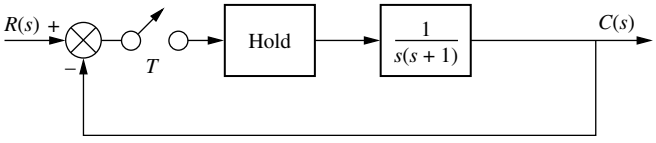


FIGURE P13.5

13. Write a MATLAB program that can be used to find the range of sampling time, T, for stability. The program will be used for systems of the type represented in Figure P13.6 and should meet the following requirements:

MATLAB ML

a. MATLAB will convert $G_1(s)$ cascaded with a sample-and-hold to $G(z)$.

b. The program will calculate the z-plane roots of the closed-loop system for a range of T and determine the value of T, if any, below which the system will be stable. MATLAB will display this value of T along with the z-plane poles of the closed-loop transfer function.

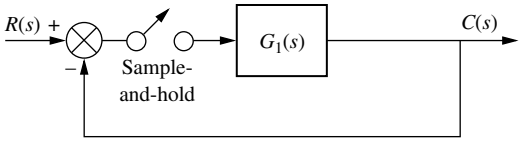


FIGURE P13.6

Test the program on

$$G_1(s) = \frac{10(s+7)}{(s+1)(s+3)(s+4)(s+5)}$$

14. Find the range of gain, K, to make the system shown in Figure P13.7 stable. [Section: 13.6]

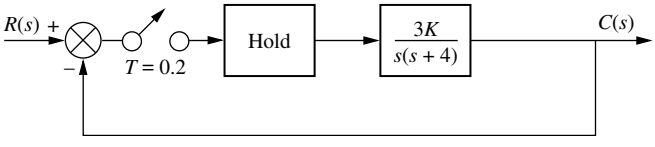


FIGURE P13.7

15. Find the static error constants and the steady-state error for each of the digital systems shown in Figure P13.8 if the inputs are [Section: 13.7]

WileyPLUS WPCS Control Solutions

a. $u(t)$

b. $tu(t)$

c. $\frac{1}{2}t^2u(t)$

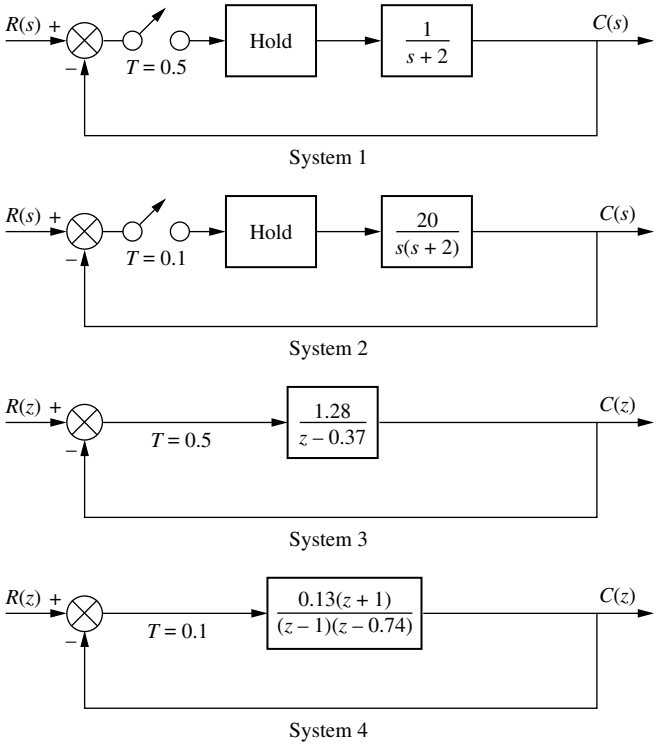


FIGURE P13.8

16. Write a MATLAB program that can be used to find K_p, K_v, and K_a for digital systems. The program will be used for systems of the type represented in Figure P13.6. Test your program for

MATLAB ML

$$G(z) = \frac{0.04406z^3 - 0.03624z^2 - 0.03284z + 0.02857}{z^4 - 3.394z^3 + 4.29z^2 - 2.393z + 0.4966}$$

where $G(z)$ is the pulse transfer function for $G_{(1)}(s)$ in cascade with the z.o.h. and $T = 0.1$ second.

17. For the digital system shown in Figure P13.6, where $G_1(s) = K/[(s+1) \times (s+4)]$, find the value of K to yield a 16.3% overshoot. Also find the range of K for stability. Let $T = 0.1$ second. [Section: 13.9]

18. Use Simulink to simulate the step response for the system of Problem 17. Set the value of gain, K, to that designed in Problem 17 for 16.3% overshoot.

Simulink SL

19. Use MATLAB's LTI Viewer to determine the peak time and settling time of the closed-loop step response for System 4 in Figure P13.8 Gui Tool GUIT

20. Write a MATLAB program that can be used to design the gain of a digital control system to meet a percent overshoot requirement. The program will be used for systems of the type represented in Figure P13.6 and meet the following requirements: MATLAB ML

a. The user will input the desired percent overshoot.

b. MATLAB will convert $G_1(s)$ cascaded with the sample-and-hold to $G(z)$.

c. MATLAB will display the root locus on the z-plane along with an overlay of the percent overshoot curve.

d. The user will click with the mouse at the intersection of the root locus and percent overshoot overlay and MATLAB will respond with the value of gain followed by a display of the step response of the closed-loop system.

Apply your program to Problem 17 and compare results.

21. For the digital system shown in Figure P13.6, where $G_1(s) = K/[s(s+1)]$, find the value of K to yield a peak time of 2 seconds if the sampling interval, T, is 0.1 second. Also, find the range of K for stability. [Section: 13.9]

22. For the digital system shown in Figure P13.6, where $G_1(s) = K/[s(s+1)(s+3)]$, find the value of K to yield a 20% overshoot if the sampling interval, T, is 0.1 second. Also, find the range of K for stability. [Section: 13.9] WileyPLUS WPCS Control Solutions

23. For the digital system shown in Figure P13.6, where $G_1(s) = K(s+2) \div [s(s+1)(s+3)]$, find the value of K to yield a settling time of 15 seconds if the sampling interval, T, is 1 second. Also, find the range of K for stability. [Section: 13.9]

24. A PID controller was designed in Example 9.5 for a continuous system with unity feedback. The system's plant was

$$G(s) = \frac{(s+8)}{(s+3)(s+6)(s+10)}$$

The designed PID controller was

$$G_c(s) = 4.6\frac{(s+55.92)(s+0.5)}{s}$$

Find the digital transfer function, $G_c(z)$, of the PID controller in order for the system to be computer controlled if the sampling interval, T, is 0.01 second. [Section: 13.10]

25. A continuous unity feedback system has a forward transfer function of WileyPLUS WPCS Control Solutions

$$G(s) = \frac{1}{s(s+5)(s+8)}$$

The system is to be computer controlled with the following specifications:

Percent overshoot: 10%
Settling time: 2 seconds
Sampling interval: 0.01 second

Design a lead compensator for the digital system to meet the specifications. [Section: 13.10]

26. Repeat Problem 25 using MATLAB. MATLAB ML

DESIGN PROBLEMS

27. a. Convert the heading control for the UFSS vehicle shown on the back endpapers (*Johnson*, *1980*) into a digitally controlled system.

b. Find the closed-loop pulse transfer function, $T(z)$, if $T = 0.1$ second.

c. Find the range of heading gain to keep the digital system stable.

28. A robot equipped to perform arc welding was discussed in Problem 45, Chapter 8. The robot was compensated by feeding back pressure and velocity signals as shown in Figure P8.13(b). Eliminating these feedback paths yields the block diagram shown in Figure P13.9 (*Hardy*, *1967*).

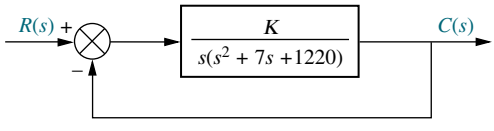


FIGURE P13.9 Simplified block diagram for robot swing motion

a. Convert the robot to a digital control system. Use a sampling time of 0.1 second.

b. Sketch the root locus.

c. Find the range of gain, K, to keep the digital system stable.

d. Repeat all previous parts using MATLAB. MATLAB ML

29. The floppy disk drive of Problem 57, Chapter 8 is to be digitally controlled. If the analog system is as shown in Figure P13.10, do the following:

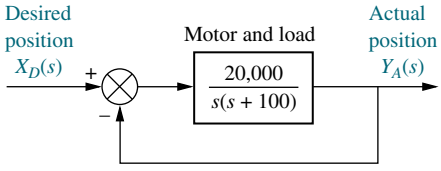


FIGURE P13.10 Simplified block diagram of a floppy disk drive

a. Convert the disk drive to a digital system. Use a sampling time of 0.01 second.

b. Find the range of digital controller gain to keep the system stable.

c. Find the value of digital controller gain to yield 15% overshoot for a digital step response.

d. Repeat all previous parts and obtain the step response for Part **c** using MATLAB. MATLAB ML

30. Scanning probe microscopes are used to visualize samples in the sub-micron dimensional range. They typically use a silica-based probe to physically track the sample topography to create a viable image. However, these devices are very sensitive to external disturbance and vibrations. An approach called *inherent disturbance suppression* tries to alleviate the disturbance problem through the addition of a laser interferometer that is used to measure the probe–sample interaction and compensate for undesired probe movements. The technique was implemented in a tapping mode atomic force microscope measuring single DNA molecules. It was shown that for a significant range of frequencies the open-loop transmission from the probe's voltage input to probe's displacement is (*Sparks*, *2006*)

$$G_1(s) = \frac{20000}{s}$$

Assuming the probe is digitally controlled in a loop, as shown in Figure P13.6, calculate the sampling period range that will result in a stable closed-loop system.

31. Problem 35, Chapter 9 described a two-tank system where the objective was to maintain a constant liquid level in one of the tanks via control of an inflow valve. Assume for this problem that the transfer function relating liquid-level output, $Y(s)$, to flow rate input $F_\theta(s)$, for the lower tank is (*Romagnoli*, *2006*)

$$G(s) = \frac{Y(s)}{F_\theta(s)} = \frac{0.0187}{s^2 + 0.237s + 0.00908}$$

Assume that the system will be controlled in closed loop by means of a digital computer system with a sample period $T = 1$ second, as shown in Figure P13.6, with $G_1(s) = KG(s)$. Use the bilinear transformation and the Routh-Hurwitz method to find the range of K that will result in a stable closed-loop system.

32. Assume that the two-tank system of Problem 31 is controlled by a digital computer in the configuration of Figure P13.6, where $G_1(s) = KG(s)$. If a sampling period of $T = 1$ second is used, do the following (*Romagnoli*, *2006*).

a. Use MATLAB to draw the root locus. MATLAB ML

b. Find the value of K that will result in a stable system with a damping factor of $\zeta = 0.7$.

c. Use the root locus of Part **b** to predict the step-response settling time, T_s, and peak time, T_p.

d. Calculate the final value of the closed-loop system to a unit step input.

e. Obtain the step response of the system using Simulink. Verify the predictions you made in Parts **c** and **d**. Simulink SL

33. In Problem 48, Chapter 9, and Problem 39, Chapter 10, we considered the radial pickup position control of a DVD player. A controller was designed and placed in cascade with the plant in a unit feedback configuration to stabilize the system. The controller was given by

$$M(s) = \frac{0.5(s + 1.63)}{s(s + 0.27)}$$

and the plant by (*Bittanti*, *2002*)

$$P(s) = \frac{0.63}{\left(1 + \frac{0.36}{305.4}s + \frac{s^2}{305.4^2}\right)\left(1 + \frac{0.04}{248.2}s + \frac{s^2}{248.2^2}\right)}$$

It is desired to replace the continuous system by an equivalent discrete system without appreciably affecting the system performance.

a. Find an appropriate sampling frequency for the discretization.

b. Using the chosen sampling frequency, translate the continuous compensator into a discrete compensator.

c. Use Simulink to simulate the continuous and discrete systems on the same graph. Simulink SL

Assume a unit step input. Are there significant differences in the system's performance?

34. In Problem 25, Chapter 11, we discussed an EVAD, a device that works in parallel with the human heart to help pump blood in patients with cardiac conditions. The device has a transfer function

$$G(s) = \frac{P_{ao}(s)}{E_m(s)} = \frac{1361}{s^2 + 69s + 70.85}$$

where $E_m(s)$ is the motor's armature voltage, and $P_{ao}(s)$ is the aortic blood pressure (*Tasch*, *1990*). Using continuous techniques, a cascaded compensator is designed in a unity feedback configuration with a transfer function

$$G_c(s) = \frac{0.5(s+1)}{s+0.05}$$

Selecting to control the device using a microcontroller, a discrete equivalent has to be found for $G_c(s)$. Do the following.

a. Find an appropriate sampling frequency for the discretization.

b. Translate the continuous compensator into a discrete compensator using the sampling frequency found in Part **a**.

c. Use Simulink to simulate the continuous and discrete systems on the same graph for a unit step input. There should be little difference between the compensated continuous and discrete systems. Simulink SL

35. In Problem 46, Chapter 9, a steam-driven turbine-governor system was implemented by a unity feedback system with a forward-path transfer function (*Khodabakhshian*, *2005*)

$$G(s) = \frac{K}{(s+0.08)(s+2)(s+5)}$$

a. Use a sampling period of $T = 0.5$ s and find a discrete equivalent for this system.

b. Use MATLAB to draw the root locus. MATLAB ML

c. Find the value of K that will result in a stable system with a damping factor of $\zeta = 0.7$.

d. Use the root locus found in Part **a** to predict the step-response settling time, T_s, and peak time, T_p.

e. Calculate the final value of the closed-loop system unit step response.

f. Obtain the step response of the system using Simulink. Verify the predictions you made in Parts **c** and **d**. Simulink SL

36. If you have not already done so, do Problem 45 in Chapter 9. In this problem, you design a PID controller for a temperature control system. Digitize your PID design and draw a flowchart from which the PID controller can be implemented. WileyPLUS WPCS Control Solutions

37. Discrete time controlled systems can exhibit unique characteristics not available in continuous controllers. For example, assuming a specific input and some conditions, it is possible to design a system to achieve steady state within one single time sample without overshoot. This scheme is well known and referred to as *deadbeat control*. We illustrate deadbeat control design with a simple example. For a more comprehensive treatment see (*Ogata*, *1987*).

Assume in Figure 13.25(a) that $G_p(s) = \dfrac{1}{s+1}$. The purpose of the design will be to find a compensator, $G_c(z)$, such that for a step input the system achieves steady state within one sample. We start by translating the system into the discrete domain to obtain the equivalent of Figure 13.25(c). The pulse transfer function, $G_p(z) = \dfrac{(1-e^{-T})z^{-1}}{1-e^{-T}z^{-1}}$, is found using Eq. (13.40), since it is assumed that the compensator will be followed by a zero-order hold. In Figure 13.25(c), the closed-loop transfer function is given by $\dfrac{C(z)}{R(z)} = T(z) = \dfrac{G_c(z)G_p(z)}{1+G_c(z)G_p(z)}$, or, solving for the compensator, we get $G_c(z) = \dfrac{1}{G_p(z)}\dfrac{T(z)}{1-T(z)}$. The desired system output is a unit step delayed by one unit sample. Thus, $C(z) = \dfrac{z}{z-1}z^{-1} = \dfrac{1}{z-1}$. Since

the input is a unit step, $R(z) = \dfrac{z}{z-1}$; the desired closed-loop transfer function is $\dfrac{C(z)}{R(z)} = T(z) = z^{-1}$, and the resulting compensator, found by direct substitution, is given by $G_c(z) = \dfrac{1}{1-e^{-T}} \dfrac{(z-e^{-T})}{z-1}$.

Assume now that the plant is given by $G_p(s) = \frac{1}{s}$, and a sampling period of $T = 0.05$ second is used.

a. Design a deadbeat compensator to reach steady state within one time sample for a step input.

b. Calculate the resulting steady-state error for a unit-slope ramp input.

c. Simulink SL Simulate your system using SIMULINK. (Hint: Following Figure 13.25, the forward path will consist of the cascading of $G_C(z)$, a zero-order hold, and $G_p(s)$.) Show that the system reaches steady state after one sample. Also verify your steady-state error ramp result.

38. LabVIEW LV Given

$$G(s) = \frac{8}{s+4}$$

Use the LabVIEW Control Design and Simulation Module to (1) convert $G(s)$ to a digital transfer function using a sampling rate of 0.25 second; and (2) plot the step responses of the discrete and the continuous transfer functions.

39. LabVIEW LV Given

$$G(z) = \frac{K(z+0.5)}{(z-0.25)(z-0.75)}$$

MATLAB ML Use the LabVIEW Control Design and Simulation Module and the MathScript RT Module to (1) obtain the value of K that will yield a damping ratio of 0.5 for the closed-loop system in Figure 13.20, where $H(z)=1$; and (2) display the step response of the closed-loop system in Figure 13.20 where $H(z) = 1$. Compare your results with those of Skill-Assessment Exercise 13.8.

PROGRESSIVE ANALYSIS AND DESIGN PROBLEMS

40. High-speed rail pantograph. Problem 21 in Chapter 1 discusses active control of a pantograph mechanism for high-speed rail systems (*O'Connor, 1997*). In Problem 79(a), Chapter 5, you found the block diagram for the active pantograph control system. In Chapter 9, you designed a PID controller to yield a settling time of 0.3 second with zero steady-state error. Assuming that the active control system is to be computer controlled, do the following:

a. Convert the PID controller designed in Problem 55, Chapter 9, to a digital controller by specifying its sampled transfer function, $G_c(z)$. Assume that the potentiometers are replaced by a keyboard, A/D converters, and unity gain transducers.

b. Draw a flowchart from which the PID controller can be implemented.

c. MATLAB ML Use MATLAB to simulate the step response of the digital active control system.

41. Control of HIV/AIDS. In Chapter 11, a continuous cascaded compensator for a unity feedback system was designed for the treatment of the HIV-infected patient treated with RTIs (*Craig, 2004*). The transfer function of the designed compensator was

$$G_c(s) = \frac{-2 \times 10^{-4}(s^2 + 0.04s + 0.0048)}{s(s+0.02)}$$

The linearized plant was given by

$$P(s) = \frac{Y(s)}{U_1(s)} = \frac{-520s - 10.3844}{s^3 + 2.6817s^2 + 0.11s + 0.0126}$$

The compensated system is overdamped with an approximate settling time of 100 seconds. This system must be discretized for practical reasons: (1) HIV patient cannot be monitored continuously and (2) medicine dosage cannot be adjusted continuously.

a. Show that a reasonable sampling period for this system is $T = 8$ days (medicine dosage will be updated on a weekly basis).

b. Use Tustin's method and $T = 8$ days to find a discrete equivalent to $G_c(S)$.

c. Simulink SL Use Simulink to simulate the continuous and discrete compensated systems for a unit step input. Plot both responses on the same graph.

42. Hybrid vehicle. MATLAB ML In Problem 7.69 (Figure P7.34), the block diagram of a cascade scheme for the speed control of an HEV (*Preitl, 2007*) was represented as a unity feedback system.

In that diagram the output of the system is the speed transducer's output voltage, $C(s) = K_{ss}V(s)$. In Part **b** of Problem 11.35, where a compensator was designed for this problem, we discussed the feasibility of achieving full pole-zero cancellation when we place a PI speed controller's zero, Z_I, on top of the uncompensated system's real pole, closest to the origin (located at -0.0163). Noting that perfect pole-zero cancellation may not be maintained, we studied a case, in which the PI-controller's zero changed by +20%, moving to -0.01304. In that case, the transfer function of the plant with a PI speed controller, which has a proportional gain $= K$, was given by:

$$G(s) = \frac{K(s + 0.6)(s + 0.01304)}{s(s + 0.0163)(s + 0.5858)}$$

Assuming that $G_1(s)$ in Figure P13.6 equals the transfer function, $G(s)$, given above for the vehicle with the speed controller:

a. Develop a MATLAB M-file that would allow you to do the following: [Hint: Refer to the M-files you developed for Problems 13 and 20 of this chapter]

(1) Convert $G_1(s)$ cascaded with a sample-and-hold to $G(z)$;

(2) Search over the range $0 < T < 5$ seconds for the largest sampling period T_{max} below which the system is stable. Calculate the z-plane roots of the closed-loop system for the whole range of the sampling time, T. Subsequently set $T = 0.75T_{max}$;

(3) Design the gain of a digital control system to meet a percent overshoot requirement, $\%OS$, allowing the user to input the value of the desired $\%OS$ and the value of the PI speed controller's proportional gain, K;

(4) Plot the step response of that digital system (in per unit, p. u., vs. time in seconds)

b. Run the M-file you developed in Part **a** and enter the values of the desired percent overshoot, $\%OS = 0$, and the PI speed controller's proportional gain, $K = 61$;

c. Select a point in the graphics window displaying the root locus, such that all poles of the closed-loop transfer function, T_z, are inside the unit circle.

d. Write the sampled-data transfer functions obtained, G_z and T_z, indicating the corresponding value of the sampling time, T, and all poles, r, of the closed-loop transfer function, T_z;

e. Plot the step response of that digital system (in per unit, p. u., vs. time in seconds) noting the following characteristics: final value, rise time, and settling time.

Cyber Exploration Laboratory

Experiment 13.1

Objective To design the gain of a digital control system to meet a transient response requirement; to simulate a digital control system to test a design; to see the effect of sampling rate upon the time response of a digital system.

Minimum Required Software Packages MATLAB, Simulink, and the Control System Toolbox

Prelab

1. Given the antenna azimuth control system shown on the front endpapers, use Configuration 2 to find the discrete transfer function of the plant. Neglect the dynamics of the power amplifier and include the preamplifier, motor, gears, and load. Assume a zero-order hold and a sampling interval of 0.01 second.

2. Using the digital plant found in Prelab 1, find the preamplifier gain required for a closed-loop digital system response with 10% overshoot and a sampling interval of 0.01 second. What is the peak time?
3. Given the antenna azimuth control system shown on the front endpapers, use Configuration 2 to find the preamplifier gain required for the continuous system to yield a closed-loop step response with 10% overshoot. Consider the open-loop system to be the preamplifier, motor, gears, and load. Neglect the dynamics of the power amplifier.

Lab

1. Verify your value of preamplifier gain found in Prelab 2 using the SISO Design Tool to generate the root locus for the digital open-loop transfer function found in Prelab 1. Use the Design Constraints capability to generate the 10% overshoot curve and place your closed-loop poles at this boundary. Obtain a plot of the root locus and the design boundary. Record the value of gain for 10% overshoot. Also, obtain a plot of the closed-loop step response using the LTI Viewer and record the values of percent overshoot and peak time. Use the same tool to find the range of gain for stability.
2. Using Simulink set up the closed-loop digital system whose plant was found in Prelab 1. Make two diagrams: one with the digital transfer function for the plant and another using the continuous transfer function for the plant preceded by a zero-order sample-and-hold. Use the same step input for both diagrams and obtain the step response of each. Measure the percent overshoot and peak time.
3. Using Simulink, set up both the digital and continuous systems calculated in Prelabs 2 and Prelab 3, respectively, to yield 10% overshoot. Build the digital system with a sample-and-hold rather than the z-transform function. Plot the step response of each system and record the percent overshoot and the peak time.
4. For one of the digital systems built in Lab 2, vary the sampling interval and record the responses for a few values of sampling interval above 0.01 second. Record sampling interval, percent overshoot, and peak time. Also, find the value of sampling interval that makes the system unstable.

Postlab

1. Make a table containing the percent overshoot, peak time, and gain for each of the following closed-loop responses: the digital system using the SISO Design Tool; the digital system using Simulink and the digital transfer functions; the digital system using Simulink and the continuous transfer functions with the zero-order sample-and-hold; and the continuous system using Simulink.
2. Using the data from Lab 4, make a table containing sampling interval, percent overshoot, and peak time. Also, state the sampling interval that makes the system unstable.
3. Compare the responses of all of the digital systems with a sampling interval of 0.01 second and the continuous system. Explain any discrepancies.
4. Compare the responses of the digital system at different sampling intervals with the continuous system. Explain the differences.
5. Draw some conclusions about the effect of sampling.

Experiment 13.2

Objective To use the various functions from the LabVIEW Control Design and Simulation Module for the analysis of digital control systems.

Minimum Required Software Packages LabVIEW with the Control Design and Simulation Module and the MathScript RT Module; MATLAB with the Control Systems Toolbox.

Prelab You are given Figure P8.28 and the parameters listed in the Prelab of Cyber Exploration Laboratory Experiment 8.2 for the open-loop NASA eight-axis ARMII (Advanced Research Manipulator II) electromechanical shoulder joint/link, actuated by an armature-controlled dc servomotor.

1. Obtain the open-loop transfer function of the shoulder joint/link, $G(s) = \dfrac{\theta_L(s)}{V_{ref}(s)}$, or use your calculation from Cyber Exploration Laboratory Experiment 8.2.
2. Use MATLAB and design a digital compensator to yield a closed-loop response with zero steady-state error and a damping ratio of 0.7. If you already have performed Cyber Exploration Laboratory Experiment 8.2, modify your M-file from that experiment. Test your design using MATLAB.

Lab Simulate your Prelab design using a Simulation Loop from the LabVIEW Control Design and Simulation Module. Plot the step response of two loops as follows: (1) a unity feedback with the forward path consisting of the continuous system transfer function preceded by a zero-order hold, and (2) a unity feedback with the forward path consisting of the equivalent discrete transfer function of your compensator in cascade with the open-loop plant.

Postlab Compare the results obtained with those from your prelab MATLAB program. Comment on time-performance specifications.

Bibliography

Astrom, K. J., and Wittenmark, B. *Computer Controlled Systems.* Prentice Hall, Upper Saddle River, NJ, 1984.

Bittanti, S., Dell'Orto, F., Di Carlo, A., and Savaresi, S. M. Notch Filtering and Multirate Control for Radial Tracking in High Speed DVD-Players. *IEEE Transactions on Consumer Electronics*, vol. 48, 2002, pp. 56–62.

Boyd, M., and Yingst, J. C. PC-Based Operator Control Station Simplifies Process, Saves Time. *Chilton's I & CS*, September 1988, pp. 99–101.

Chassaing, R. *Digital Signal Processing.* Wiley, New York, 1999.

Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Transactions on Education*, vol. 47, no. 1, February 2004, pp. 65–73.

Craig, J. J. *Introduction to Robotics. Mechanics and Control*, 3d ed. Prentice Hall, Upper Saddle River, NJ, 2005.

Hardy, H. L. Multi-Loop Servo Controls Programmed Robot. *Instruments and Control Systems*, June 1967, pp. 105–111.

Hostetter, G. H. *Digital Control System Design*. Holt, Rinehart & Winston, New York, 1988.

Johnson, H. et al. *Unmanned Free-Swimming Submersible* (*UFSS*) *System Description*. NRL Memorandum Report 4393. Naval Research Laboratory, Washington, D.C., 1980.

Katz, P. *Digital Control Using Microprocessors.* Prentice Hall, Upper Saddle River, NJ, 1981.

Khodabakhshian, A., and Golbon, N. Design of a New Load Frequency PID Controller Using QFT. *Proceedings of the 13th Mediterranean Conference on Control and Automation*, 2005, pp. 970–975.

Kuo, B. C. *Automatic Control Systems*, 7th ed. Prentice Hall, Upper Saddle River, NJ, 1995.

Kuo, B. C. *Digital Control Systems.* Holt, Rinehart & Winston, New York, 1980.

Nyzen, R. J. *Analysis and Control of an Eight-Degree-of-Freedom Manipulator*, Ohio University Masters Thesis, Mechanical Engineering, Dr. Robert L. Williams II, Advisor, August 1999.

O'Connor, D. N., Eppinger, S. D., Seering, W. P., and Wormly, D. N. Active Control of a High-Speed Pantograph. *Journal of Dynamic Systems, Measurements, and Control*, vol. 119, March 1997, pp. 1–4.

Ogata, K. *Discrete-Time Control Systems.* Prentice Hall, Upper Saddle River, NJ, 1987.

Phillips, C. L., and Nagle, H. T., Jr., *Digital Control System Analysis and Design*. Prentice Hall, Upper Saddle River, NJ, 1984.

Preitl, Z., Bauer, P., and Bokor, J. A Simple Control Solution for Traction Motor Used in Hybrid Vehicles. *4th International Symposium on Applied Computational Intelligence and Informatics.* IEEE, 2007.

Romagnoli, J. A., and Palazoglu, A. *Introduction to Process Control*. CRC Press, Boca Raton, FL, 2006.

Smith, C. L. *Digital Computer Process Control*. Intext Educational Publishers, NY, 1972.

Sparks, A. W., and Manalis, S. R. Atomic Force Microscopy with Inherent Disturbance Suppression for Nanostructure Imaging. *Nanotechnology*, vol. 17, 2006, pp. 1574–1579.

Tasch, U., Koontz, J. W., Ignatoski, M. A., and Geselowitz, D. B. An Adaptive Aortic Pressure Observer for the Penn State Electric Ventricular Assist Device. *IEEE Transactions on Biomedical Engineering*, vol. 37, 1990, pp. 374–383.

Tou, J. *Digital and Sampled-Data Control Systems.* McGraw-Hill, New York, 1959.

Williams, R. L. II. Local Performance Optimization for a Class of Redundant Eight-Degree-of-Freedom Manipulators. *NASA Technical Paper 3417*, NASA Langley Research Center, Hampton VA, March 1994.

This page intentionally left blank

Appendix A: List of Symbols

$\%OS$	Percent overshoot
A	Ampere—unit of electrical current
$\mathbf{A}$	System matrix for state-space representation
a_m	Motor time constant
β	Mechanical rotational coefficient of viscous friction in N-m-s/rad
$\mathbf{B}$	Input matrix for state-space representation
C	Electrical capacitance in farads
$\mathbf{C}$	Output matrix for state-space representation
$C(s)$	Laplace transform of the output of a system
$c(t)$	Output of a system
$\mathbf{C_M}$	Controllability matrix
D	Mechanical rotational coefficient of viscous friction in N-m-s/rad
$\mathbf{D}$	Feedforward matrix for state-space representation
D_a	Motor armature coefficient of viscous damping in N-m-s/rad
D_m	Total coefficient of viscous friction at the armature of a motor, including armature coefficient of viscous friction and reflected load coefficient of viscous friction in N-m-s/rad
E	Energy
$E(s)$	Laplace transform of the error
$e(t)$	Error; electrical voltage
$E_a(s)$	Laplace transform of the motor armature input voltage; Laplace transform of the actuating signal
$e_a(t)$	Motor armature input voltage; actuating signal
F	Farad—unit of electrical capacitance
$F(s)$	Laplace transform of $f(t)$
$f(t)$	Mechanical force in newtons; general time function
f_v	Mechanical translational coefficient of viscous friction
g	Acceleration due to gravity
G	Electrical conductance in mhos
$G(s)$	Forward-path transfer function
$G_c(s)$	Compensator transfer function
$G_c(z)$	Sampled transfer function for a compensator
G_M	Gain margin
$G_p(z)$	Sampled transfer function for a plant

H	Henry—unit of electrical inductance
$H(s)$	Feedback-path transfer function
I	Identity matrix
$i(t)$	Electrical current in amperes
J	Moment of inertia in kg-m^2
J_a	Motor armature moment of inertia in kg-m^2
J_m	Total moment of inertia at the armature of a motor, including armature moment of inertia and reflected load moment of inertia in kg-m^2
K	Controller gain matrix
K	Mechanical translational spring constant in N/m or rotational spring constant in N-m/rad; amplifier gain; residue
k	Controller feedback gain; running index
K_a	Acceleration constant
K_b	Back emf constant in V/rad/s
K_f	Feedback gain
kg	Kilogram = newton seconds^2/meter—unit of mass
kg-m^2	Kilogram meters^2 = newton-meters seconds^2/radian—unit of moment of inertia
K_m	Motor gain
K_p	Position constant
K_t	Motor torque constant relating developed torque to armature current in N-m/A
K_v	Velocity constant
L	Electrical inductance in henries
L	Observer gain matrix
l	Observer feedback gain
M	Mass in kilograms; slope of the root locus asymptotes
m	Meter—unit of mechanical translational displacement
$M(\omega)$	Magnitude of a sinusoidal response
m/s	Meters/second—unit of mechanical translational velocity
M_P	Peak magnitude of the sinusoidal magnitude response
N	Newton—unit of mechanical translational force in kilogram meters/second^2
N-s/m	Newton-seconds/meter—unit of mechanical translational coefficient of viscous friction
n	System type
N/m	Newton/meter—unit of mechanical translational spring constant
N-m	Newton-meter—unit of mechanical torque
N-m-s/rad	Newton-meter-seconds/radian—unit of mechanical rotational coefficient of viscous friction
N-m/A	Newton-meter/ampere—unit of motor torque constant
N-m/rad	Newton-meter/radian—unit of mechanical rotational spring constant
$\mathbf{O_M}$	Observability matrix
P	Similarity transformation matrix
p_c	Compensator pole
Q	Coulomb—unit of electrical charge
$q(t)$	Electrical charge in coulombs

R	Electrical resistance in ohms
$R(s)$	Laplace transform of the input to a system
r	Nonlinear electrical resistance
$r(t)$	Input to a system
R_a	Motor armature resistance in ohms
rad	Radian—unit of angular displacement
rad/s	Radian/second—unit of angular velocity
s	Second—unit of time
s	Complex variable for the Laplace transform
$S_{F:P}$	Sensitivity of F to a fractional change in P
T	Time constant; sampling interval for digital signals
$T(s)$	Closed-loop transfer function; Laplace transform of mechanical torque
$T(t)$	Mechanical torque in N-m
$T_m(t)$	Torque at the armature developed by a motor in N-m
$T_m(s)$	Laplace transform of the torque at the armature developed by a motor
T_p	Peak time in seconds
T_r	Rise time in seconds
T_s	Settling time in seconds
T_w	Pulse width in seconds
$\mathbf{u}$	Input or control vector for state-space representation
u	Input control signal for state-space representation
$u(t)$	Unit step input
V-s/rad	Volt-seconds/radian—unit of motor back emf constant
$v(t)$	Mechanical translation velocity in m/s; electrical voltage
$v_b(t)$	Motor back emf in volts
$v_e(t)$	Error voltage
$v_p(t)$	Power amplifier input in volts
$\mathbf{x}$	State vector for state-space representation
$x(t)$	Mechanical translation displacement in meters; a state variable
$\dot{x}$	Time derivative of a state variable
$\dot{\mathbf{x}}$	Time derivative of the state vector
$\mathbf{y}$	Output vector for state-space representation
$y(t)$	Output scalar for state-space representation
z	Complex variable for the z-transform
z_c	Compensator zero
α	Pole-scaling factor for a lag compensator, where $\alpha > 1$; angle of attack
β	Pole-scaling factor for a lead compensator, where $\beta < 1$
γ	Pole-scaling factor for a lag-lead compensator, where $\gamma > 1$
δ	Thrust angle
ζ	Damping ratio
θ	Angle of a vector with the positive extension of the real axis
$\theta(t)$	Angular displacement
θ_a	Angle of a root locus asymptote with the positive extension of the real axis

θ_c	Angular contribution of a compensator on the *s*-plane
$\theta_m(t)$	Angular displacement of the armature of a motor
λ	Eigenvalue of a square matrix
σ	Real part of the Laplace transform variable, *s*
σ_a	Real-axis intercept of the root locus asymptotes
Φ_M	Phase margin
$\Phi(t)$	State transition matrix
ϕ	Sinusoidal phase angle; body angle
ϕ_c	Sinusoidal phase angle of a compensator
ϕ_{max}	Maximum sinusoidal phase angle
Ω	Ohm—unit of electrical resistance
$\mho$	Mho—unit of electrical conductance
ω	Imaginary part of the Laplace transform variable, *s*
$\omega(t)$	Angular velocity in rad/s
ω_{BW}	Bandwidth in rad/s
ω_d	Damped frequency of oscillation in rad/s
$\omega\phi_M$	Phase-margin frequency in radians
ω_{G_M}	Gain-margin frequency in radians
ω_n	Natural frequency in rad/s
ω_p	Peak-magnitude frequency of the magnitude frequency response in rad/s

Appendix B: MATLAB Tutorial

B.1 Introduction

MATLAB is a high-level technical computing environment suitable for solving scientific and engineering problems. When used with routines from its companion software, the Control System Toolbox, MATLAB can be used to analyze and design control systems problems such as those covered in this textbook. MATLAB and the Control System Toolbox are commercial software products available from The Math-Works, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098. Phone: (508) 647-7000. Email: info@mathworks.com. URL: www.mathworks.com.

The MATLAB examples in this tutorial consist of solved problems that demonstrate the application of MATLAB to the analysis and design of control systems. Many problems were taken from examples in the text (identified with a **MATLAB** icon) that were solved without MATLAB. A Command Summary at the end of this appendix lists key MATLAB statements and their descriptions.

The code in this tutorial is also available in the Control Systems Engineering Toolbox folder at www.wiley.com/college/nise and at www.mathworks.com/matlabcentral/fileexchange. You should have MATLAB 2009b and the Control System Toolbox Version 8.4 installed on your machine to execute this appendix's code in the Control Systems Engineering Toolbox Version 6.

To run the M-files, first be sure the files are either added to the search path in **Set Path** . . . under the **File** menu or appear in the **Current Folder** window, which is part of the **MATLAB** window. To see the computer responses after installing the M-files, run each problem by typing the M-file name, such as ch2p1, after the prompt (>>) in the **Command Window.** You may also run the files by right-clicking the file name, if it appears in the **Current Folder** window, and select **Run File.**

To view all or part of the M-file in the **Command Window,** enter "type <file name>" or "help <file name>," respectively, after the prompt. You may also view and make changes to the M-file by double-clicking the file in the **Current Folder** window. This action brings up the editor. After editing, be sure to save the revised file before executing.

If you do not have the Control Systems Engineering Toolbox M-files, you can create your own M-files by typing the code for each problem in this appendix into a separate M-file (there is no need to type the final pause statement or comments), and naming each M-file with a .m extension, as in ch2p1.m. You can also type the code for more than one problem into an M-file, including the pause command, and name the M-file with the .m extension. You can then call the file from the **Command Window,** and continue past the pause statements to the next problem by pressing any key.

By its nature, this appendix cannot cover all the background and details necessary for a complete understanding of MATLAB. For further details, you are referred to other sources, including MATLAB reference manuals and instructions specific to your particular computer. The bibliography at the end of this appendix provides a partial listing of references. This appendix should give you enough information to be able to apply MATLAB to the analysis and design problems covered in this book.

The code contained in this appendix and in the Control Systems Engineering Toolbox was developed on a PC using MATLAB Version 7.9 and the Control System Toolbox Version 8.4. The code will also run on workstations that support MATLAB. Consult the MATLAB *Installation Guide* for your platform for minimum system hardware requirements.

B.2 MATLAB Examples

Chapter 2: Modeling in the Frequency Domain

ch2p1 Bit strings will be used to identify parts of this tutorial on the computer output. Bit strings are represented by the text enclosed in apostrophes, such as `'ab'`. Comments begin with `%` and are ignored by MATLAB. Numbers are entered without any other characters. Arithmetic can be performed using the proper arithmetic operator. Numbers can be assigned using a left-hand argument and an equals sign. Finally, we can find the magnitude and angle of a complex number, Q using `abs` (Q) and `angle` (Q), respectively.

```
'(ch2p1)'                          % Display label.
'How are you?'                     % Display string.
-3.96                              % Display scalar number -3.96.
-4 + 7i                            % Display complex number -4+7i.
-5-6j                              % Display complex number -5-6 j.
(-4+7i)+(-5-6i)                    % Add two complex numbers and
                                   % display sum.
(-4+7j)*(-5-6j)                    % Multiply two complex numbers and
                                   % display product.
M=5                                % Assign 5 to M and display.
N=6                                % Assign 6 to N and display.
P=M+N                              % Assign M+N to P and display.
Q=3+4j                             % Define complex number, Q.
MagQ=abs(Q)                        % Find magnitude of Q.
ThetaQ=(180/pi)*angle(Q)           % Find the angle of Q in degrees.
pause
```

ch2p2 Polynomials in s can be represented as row vectors containing the coefficients. Thus $P_1 = s^3 + 7s^2 - 3s + 23$ can be represented by the vector shown below with elements separated by a space or comma. Bit strings can be used to identify each section of this tutorial.

```
'(ch2p2)'                          % Display label.
P1=[17-323]                        % Store polynomial s^3 + 7s^2 -3s+
                                   % 23 as P1 and display.
pause
```

ch2p3 Running the previous statements causes MATLAB to display the results. Ending the command with a semicolon suppresses the display. Typing an expression without a left-hand assignment and without a semicolon causes the expression to be evaluated and the result displayed. Enter `P2` in the **MATLAB Command Window** after execution.

```
'(ch2p3)'                          % Display label.
P2 =[3 5 7 8];                     % Assign 3s^3 + 5s^2 +7s + 8 to P2
                                   % without displaying.
3*5                                % Evaluate 3*5 and display result.
pause
```

ch2p4 An $F(s)$ in factored form can be represented in polynomial form. Thus $P_3 = (s+2)(s+5)(s+6)$ can be transformed into a polynomial using `poly (V)`, where `V` is a row vector containing the roots of the polynomial and `poly(V)` forms the coefficients of the polynomial.

```
'(ch2p4)'                          % Display label.
P3=poly([-2 -5 -6])                % Store polynomial
                                   % (s+2) (s+5)(s+6) as P3 and
                                   % display the coefficients.
pause
```

ch2p5 We can find roots of polynomials using the `roots (V)` command. The roots are returned as a column vector. For example, find the roots of $5s^4 + 7s^3 + 9s^2 - 3s + 2 = 0$.

```
'(ch2p5)'                          % Display label.
P4=[5 7 9 -3 2]                    % Form 5s^4+7s^3+9s^2-3s+2 and
                                   % display.
rootsP4=roots(P4)                  % Find roots of 5s^4+7s^3+9s^2
                                   %-3s+2,
                                   % assign to rootsP4, and display.
pause
```

ch2p6 Polynomials can be multiplied together using the `conv(a,b)` command (standing for convolve). Thus, $P_5 = (s^3 + 7s^2 + 10s + 9)(s^4 - 3s^3 + 6s^2 + 2s + 1)$ is generated as follows:

```
'(ch2p6)'                          % Display label.
P5=conv([1 7 1 0 9],[1 -3 6 2 1])  % Form (s^3+7s^2+10s+9)(s^4-
                                   % 3s^3+6s^2+2s+l), assign to P5,
                                   % and display.
pause
```

ch2p7 The partial-fraction expansion for $F(s) = b(s)/a(s)$ can be found using the `[K, p, k] = residue (b, a)` command (`K` = residue; `p` = roots of denominator; `k` = direct quotient, which is found by dividing polynomials prior to performing a partial-fraction expansion). We expand $F(s) = (7s^2 + 9s + 12)/[s(s + 7)(s^2 + 10s + 100)]$ as an example. Using the results from MATLAB yields: $F(s) = [(0.2554 - 0.3382i) / (s + 5.0000 - 8.6603i)] + [(0.2554 + 0.3382i) / (s + 5.0000 + 8.6603i)] - [0.5280/ (s + 7)] + [0.0171/s]$.

```
'(ch2p7)'                                % Display label.
numf=[7 9 12]                            % Define numerator of F(s).
denf=conv(poly([0-7]),[1 10 100]);       % Define denominator of F(s).
[K, p, k]=residue (numf ,denf)           % Find residues and assign to K;
                                         % find roots of denominator and
                                         % assign to p; find
                                         % constant and assign to k.
pause
```

ch2p8 (Example 2.3) Let us do Example 2.3 in the book using MATLAB.

```
'(ch2p8) Example 2.3'                    % Display label.
numy=32;                                 % Define numerator.
deny=poly([0 -4 -8]);                    % Define denominator.
[r, p, k]=residue (numy, deny)           % Calculate residues, poles, and
                                         % direct quotient.
pause
```

ch2p9 Creating Transfer Functions

Vector Method, Polynomial Form

A transfer function can be expressed as a numerator polynomial divided by a denominator polynomial, that is, $F(s) = N(s)/D(s)$. The numerator, $N(s)$, is represented by a row vector, `numf`, that contains the coefficients of $N(s)$. Similarly, the denominator, $D(s)$, is represented by a row vector, `denf`, that contains the coefficients of $D(s)$. We form $F(s)$ with the command, `F=tf (numf, denf)`. `F` is called a linear time-invariant (LTI) object. This object, or transfer function, can be used as an entity in other operations, such as addition or multiplication. We demonstrate with $F(s) = 150(s^2 + 2s + 7)/[s(s^2 + 5s + 4)]$. Notice after executing the `tf` command, MATLAB prints the transfer function.

Vector Method, Factored Form We also can create LTI transfer functions if the numerator and denominator are expressed in factored form. We do this by using row vectors containing the roots of the numerator and denominator. Thus $G(s) = K^*N(s)/D(s)$ can be expressed as an LTI object using the command, `G=zpk (numg, deng, K)`, where `numg` is a row vector containing the roots of $N(s)$ and `deng` is a row vector containing the roots of $D(s)$. The expression `zpk` stands for zeros (roots of the numerator), poles (roots of the denominator), and gain, K. We demonstrate with $G(s) = 20(s + 2)(s + 4)/[(s + 7)(s + 8)(s + 9)]$. Notice after executing the `zpk` command, MATLAB prints the transfer function.

Rational Expression in s Method, Polynomial Form (Requires Control System Toolbox 8.4) This method allows you to type the transfer function as you normally would write it. The statement `s=tf('s')` must precede the transfer function if you wish to create an LTI transfer function in polynomial form equivalent to using `G=tf (numg, deng)`.

Rational Expression in s Method, Factored Form (Requires Control System Toolbox 8.4) This method allows you to type the transfer function as you normally would write it. The statement `s=zpk ('s')` must precede the transfer function if you wish to create an LTI transfer function in factored form equivalent to using `G=zpk (numg, deng, K)`.

For both rational expression methods the transfer function can be typed in any form regardless of whether `s=tf('s')` or `s=zpk('s')` is used. The difference is in the created LTI transfer function. We use the same examples above to demonstrate the rational expression in s methods.

```
'(ch2p9)'                          % Display label.
'Vector Method, Polynomial Form'   % Display label.
numf=150*[1 2 7]                   % Store 150 (s^2+2s+7) in numf and
                                   % display.
denf=[1 5 4 0]                     % Store s (s+1) (s+4) in denf and
                                   % display.
'F(s)'                             % Display label.
F=tf (numf, denf)                  % Form F(s) and display.
clear                              % Clear previous variables from
                                   % workspace.
'Vector Method, Factored Form'     % Display label.
numg=[-2 -4]                       % Store (s+2)(s+4) in numg and
                                   % display.
deng=[-7 -8 -9]                    % Store (s+7) (s+8) (s+9) in deng
                                   % and display.
K=20                               % Define K.
'G(s)'                             % Display label.
G=zpk(numg,deng,K)                 % Form G(s) and display.
clear                              % Clear previous variables from
                                   % workspace.
'Rational Expression Method, Polynomial Form'
                                   % Display label.
s=tf ('s')                         % Define's' as an LTI object in
                                   % polynomial form.
F=150*(s^2+2*s+7)/[s*(s^2+...      % Form F (s) as an LTI transfer
5*s+4)]                            % function in polynomial form.
G=20*(s+2)*(s+4)=[(s+7)*...        % Form G(s) as an LTI transfer
(s+8)*(s+9)]                       % function in polynomial form.
clear                              % Clear previous variables from
                                   % workspace.
'Rational Expression Method, Factored Form'
                                   % Display label.
s=zpk('s')                         % Define's' as an LTI object in
                                   % factored form.
F=150*(s^2+2*s+7)/[s*(s^2+5*s+4)]
                                   % Form F (s) as an LTI transfer
                                   % function in factored form.
G=20*(s+2)*(s+4)/[(s+7)*(s+8)*(s+9)]
                                   % Form G(s) as an LTI transfer
                                   % function in factored form.
pause
```

ch2p10 Transfer function numerator and denominator vectors can be converted between polynomial form containing the coefficients and factored form containing the roots. The MATLAB function, `tf 2 zp (numtf, dentf)`, converts the numerator and denominator from coefficients to roots. The results are in the form of column vectors. We demonstrate this with $F(s) = (10s^2 + 40s + 60)/(s^3 + 4s^2 + 5s + 7)$. The MATLAB function, `zp2tf(numzp,denzp,K)`, converts the numerator and denominator from roots to coefficients. The arguments `numzp` and `denzp` must be

column vectors. In the demonstration that follows, apostrophes signify transpose. We demonstrate the conversion from roots to coefficients with $G(s) = 10(s+2)(s+4)/[s(s+3)(s+5)]$.

```
'(ch2p10)'                          % Display label.
'Coefficients for F(s)'             % Display label.
numftf=[10 4 0 60]                  % Form numerator of F(s)=
                                    % (10s^2+40s+60)/(s^3+4s^2+5s
                                    % +7).
denftf=[1 4 5 7]                    % Form denominator of F(s)=
                                    % (10s^2+40s+60)/(s^3+4s^2+5s
                                    % +7).
'Roots for F(s)'                    % Display label.
[numfzp,denfzp]=tf2zp(numftf,denftf)
                                    % Convert F(s) to factored form.
'Roots for G(s)'                    % Display label.
numgzp=[-2 -4]                      % Form numerator of
K=10                                % G(s)=10(s+2)(s+4)=[s(s+3)
                                    % (s+5)].
dengzp=[0 -3 -5]                    % Form denominator of
                                    % G(s)=10(s+2)(s+4)/[s(s+3)(s+5)].
'Coefficients for G(s)'             % Display label.
[numgtf, dengtf]=zp2tf(numgzp', dengzp', K)
                                    % Convert G(s) to polynomial form.
pause
```

ch2p11 LTI models can also be converted between polynomial and factored forms. MATLAB commands `tf` and `zpk` are also used for the conversion between LTI models. If a transfer function, *Fzpk*(*s*), is expressed as factors in the numerator and denominator, then `tf (Fzpk)` converts *Fzpk*(*s*) to a transfer function expressed as coefficients in the numerator and denominator. Similarly, if a transfer function, *Ftf*(*s*), is expressed as coefficients in the numerator and denominator, then `zpk(Ftf)` converts *Ftf*(*s*) to a transfer function expressed as factors in the numerator and denominator. The following example demonstrates the concepts.

```
'(ch2p11)'                          % Display label.
'Fzpk1(s)'                          % Display label.
Fzpk1=zpk([-2-4],[0-3-5],10)        % Form Fzpk1(s)=
                                    % 10(s+2)(s+4)/[s(s+3)(s+5)].
'Ftf1'                              % Display label.
Ftf1=tf(Fzpk1)                      % Convert Fzpk1(s) to
                                    % coefficients form.
'Ftf2'                              % Display label.
Ftf2=tf([10 40 60],[1 4 5 7])       % Form Ftf2(s)=
                                    % (10s^2+40s+60)/(s^3+4s^2+5s
                                    % +7).
'Fzpk2'                             % Display label.
Fzpk2=zpk(Ftf2)                     % Convert Ftf2(s) to
                                    % factored form.
pause
```

ch2p12 Functions of time can be easily plotted using MATLAB's `plot (X, Y, S)`, where `X` is the independent variable, `Y` is the dependent variable, and `S` is a character string describing the plot's color, marker, and line characteristic. Type

`HELP PLOT` in the **Command Window** to see a list of choices for `S`. Multiple plots also can be obtained using `plot (X1, Y1, S1, X2, Y2, S2, X3, Y3, S3, . . . )`. In the following example we plot on the same graph $\sin(5t)$ in red and $\cos(5t)$ in green for $t = 0$ to 10 seconds in 0.01 second increments. Time is specified as `t=start: increment: final`.

```
'(ch2p12)'                          % Display label.
t=0:0.01:10;                        % Specify time range and increment.
f1=cos(5*t);                        % Specify f1 to be cos(5t).
f2=sin(5*t);                        % Specify f2 to be sin (5t).
plot(t,f1,'r',t,f2,'g')             % Plot f1 in red and f2 in green.
pause
```

Chapter 3: Modeling in the Time Domain

ch3p1 The square system matrix, $\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -9 & -8 & -7 \end{bmatrix}$ is written with a space or comma separating the elements of each row. The next row is indicated with a semicolon or carriage return. The entire matrix is then enclosed in a pair of square brackets.

```
'(ch3p1)'                           % Display label.
A=[0 1 0;0 0 1; -9 -8 -7]           % Represent A.
'or'
A=[0 1 0                            % Represent A.
0 0 1
-9 -8 -7]
pause
```

ch3p2 A row vector, such as the output matrix **C**, can be represented with elements separated by spaces or commas and enclosed in square brackets. A column vector, such as input matrix **B**, can be written as elements separated by semicolons or carriage returns, or as the transpose (') of a row vector.

```
'(ch3p2)'                           % Display label.
C=[2 3 4]                           % Represent row vector C.
B=[7 ; 8 ; 9]                       % Represent column vector B.
'or'
B=[7                                % Represent column vector B.
8
9]
'or'
B=[7 8 9]'                          % Represent column vector B.
pause
```

ch3p3 The state-space representation consists of specifying the **A**, **B**, **C**, and **D** matrices followed by the creation of an LTI state-space object using the MATLAB command, `ss(A,B,C,D)`. Hence, for the matrices in (ch3p1) and (ch3p2), the state-space representation would be:

```
'(ch3p3)'                          % Display label.
A=[0 1 0;0 0 1; -9 -8 -7]          % Represent A.
B=[7;8;9];                         % Represent column vector B.
C=[2 3 4];                         % Represent row vector C.
D=0;                               % Represent D.
F=ss(A,B,C,D)                      % Create an LTI object and display.
```

ch3p4 (Example 3.4) Transfer functions represented either by numerator and denominator or an LTI object can be converted to state space. For numerator and denominator representation, the conversion can be implemented using `[A, B, C, D] =tf2ss (num, den)`. The **A** matrix is returned in a form called the controller canonical form, which will be explained in Chapter 5 in the text. To obtain the phase-variable form, `[Ap, Bp, Cp, Dp]`, we perform the following operations: `Ap=inv (P) *A*P; Bp=inv (P)*B; Cp=C*P, Dp=D`, where **P** is a matrix with 1s along the anti-diagonal and 0s elsewhere. These transformations will be explained in Chapter 5. The command `inv (X)` finds the inverse of a square matrix. The symbol * signifies multiplication. For systems represented as LTI objects, the command `ss (F)`, where `F` is an LTI transfer-function object, can be used to convert `F` to a state-space object. Let us look at Example 3.4 in the text. For the numerator-denominator representation, notice that the MATLAB response associates the gain, 24, with the vector **C** rather than the vector **B** as in the example in the text. Both representations are equivalent. For the LTI transfer-function object, the conversion to state space does not yield the phase-variable form. The result is a balanced model that improves the accuracy of calculating eigenvalues, which are covered in Chapter 4. Since `ss (F)` does not yield familiar forms of the state equations (nor is it possible to easily convert to familiar forms), we will have limited use for that transformation at this time.

```
' (ch3p4) Example 3.4'             % Display label.
'Numerator-denominator representation conversion'
                                   % Display label.
'Controller canonical form'        % Display label.
num=24 ;                           % Define numerator of
                                   % G(s)=C(s)/R(s).
den=[1 9 26 24];                   % Define denominator of G(s).
[A,B,C,D]=tf2ss(num,den)           % Convert G(s) to controller
                                   % canonical form, store matrices
                                   % A, B, C, D, and display.
'Phase-variable form'              % Display label.
P=[0 0 1;0 10;10 0];               % Form transformation matrix.
Ap=inv(P)*A*P                      % Form A matrix, phase-variable
                                   % form.
Bp=inv(P)*B                        % Form B vector, phase-variable
                                   % form.
Cp=C*P                             % Form C vector, phase-variable
                                   % form.
Dp=D                               % Form D phase-variable form.
'LTI object representation'        % Display label.
T=tf(num,den)                      % Represent T(s)=24/(s^3+9s^2 +
                                   % 26s+24) as an LTI transfer-
                                   % function object.
Tss=ss(T)                          % Convert T(s) to state space.
pause
```

ch3p5 State-space representations can be converted to transfer functions represented by a numerator and a denominator using `[num,den]=ss2tf(A,B,C,D,iu)`, where `iu` is the input number for multiple-input systems. For single-input, single-output systems `iu=1`. For an LTI state-space system, `Tss`, the conversion can be implemented using `Ttf=tf(Tss)` to yield the transfer function in polynomial form or `Tzpk=zpk (Tss)` to yield the transfer function in factored form. For example, the transfer function represented by the matrices described in (ch3p3) can be found as follows:

```
'(ch3p5)'                          % Display label.
         'Non LTI'                 % Display label.
A=[0 1 0;0 0 1;-9 -8 -7];          % Represent A.
B=[7;8;9];                         % Represent B.
C=[2 3 4]                          % RepresentC.
D=0;                               % Represent D.
'Ttf (s)'                          % Display label.
[num,den]=ss2tf (A,B,C,D,1)        % Convert state-space
                                   % representation to a
                                   % transfer function represented as
                                   % a numerator and denominator in
                                   % polynomial form, G(s)=num/den,
                                   % and display num and den.
         'LTI'                     % Display label.
Tss=ss(A,B,C,D)                    % Form LTI state-space model.
'Polynomial form, Ttf(s)'          % Display label.
Ttf=tf (Tss)                       % Transform from state space to
                                   % transfer function in polynomial
                                   % form.
'Factored form, Tzpk(s)'           % Display label.
Tzpk=zpk(Tss)                      % Transform from state space to
                                   % transfer function in factored
                                   % form.
pause
```

Chapter 4: Time Response

ch4p1 (Example 4.6) We can use MATLAB to calculate characteristics of a second-order system, such as damping ratio, ζ; natural frequency, ω_n; percent overshoot, $\%OS$ (pos); settling time, T_s; and peak time, T_p. Let us look at Example 4.6 in the text.

```
'(ch4p1) Example 4.6'              % Display label.
p1=[1 3+7*i];                      % Define polynomial containing
                                   % first pole.
p2=[1 3 -7*i];                     % Define polynomial containing
                                   % second pole.
deng=conv(p1 ,p2);                 % Multiply the two polynomials to
                                   % find the 2nd order polynomial,
                                   % as^2+bs+c.
omegan=sqrt (deng (3) /deng (1))   % Calculate the natural frequency,
                                   % sqrt(c/a).
zeta=(deng(2)/deng(1))/(2*omegan)
                                   % Calculate damping ratio,
                                   % ((b/a)/2*wn).
Ts=4/(zeta*omegan)                 % Calculate settling time,
                                   % (4/z*wn).
```

```
Tp=pi/(omegan*sqrt(1-zeta^2))      % Calculate peak time,
                                   % pi/wn*sqrt(1-z^2).
pos=100*exp(-zeta*pi/sqrt(1-zeta^2))
                                   % Calculate percent overshoot
                                   % (100*e^(-z*pi/sqrt(1-z^2)).
pause
```

ch4p2 (Example 4.8) We can use MATLAB to obtain system step responses. These responses are particularly valuable when the system is not a pure two-pole system and has additional poles or zeros. We can obtain a plot of the step response of a transfer function, $T(s)$ =`num/den`, using the command `step(T)`, where `T` is an LTI transfer-function object. Multiple plots also can be obtained using `step(T1, T2,...)`

Information about the plots obtained with `step(T)` can be found by left-clicking the mouse on the curve. You can find the curve's label as well as the coordinates of the point on which you clicked. Right-clicking away from a curve brings up a menu. From this menu you can select (1) system responses to be displayed and (2) response characteristics to be displayed, such as peak response. When selected, a dot appears on the curve at the appropriate point. Let your mouse rest on the point to read the value of the characteristic. You may also select (3) choice for grid on or off, (4) choice to normalize the curve, and (5) properties, such as labels, limits, units, style, and characteristics.

If we add the left-hand side, `[y, t]=step (T)`, we create vectors containing the plot's points, where `y` is the output vector and `t` is the time vector. For this case, a plot is not made until the `plot (t, y)` command is given, where we assume we want to plot the output `(y)` versus time `(t)`. We can label the plot, the x-axis, and the y-axis with `title('ab') , xlabel('ab') , and ylabel('ab')`, respectively. The command `clf` clears the graph prior to plotting. Finally, text can be placed anywhere on the graph using the command `text (X, Y, 'text')`, where `(X, Y)` are the graph coordinates where `'text'` will be displayed. Let us look at Example 4.8 in the text.

```
'(ch4p2) Example 4.8'              % Display label.
'Test Run'                         % Display label.
clf                                % Clear graph.
numt1=[24.542];                    % Define numerator of T1.
dent1=[1 4 24.542];                % Define denominator of T1.
'T1(s)'                            % Display label.
T1=tf(numt1,dent1)                 % Create and display T1(s).
step(T1)                           % Run a demonstration step response
                                   % plot
title ('Test Run of T1 (s)')       % Add title to graph.
pause
'Complete Run'                     % Display label.
[y1,t1]=step(T1);                  % Run step response of T1 and
                                   % collect points.
numt2=[245.42];                    % Define numerator of T2.
p1=[1 10];                         % Define (s+10) in denominator
                                   % of T2.
p2=[1 4 24.542];                   % Define (s^2+4s+24.542) in
                                   % denominator of T2.
dent2=conv(p1,p2);                 % Multiply (s + 10)(s^2+4s+24.542)
                                   % for denominator of T2.
'T2(s)'                            % Display label.
T2=tf(numt2, dent2)                % Create and display T2.
```

```
[y2,t2]=step(T2);                    % Run step response of T2 and
                                     % collect points.
numt3=[73.626];                      % Define numerator of T3.
p3=[1 3];                            % Define (s+3) in denominator
                                     % of T3.
dent3=conv(p3,p2);                   % Multiply (s+3) (s^2+4s+24.542)
                                     % for denominator of T3.
'T3(s)'                              % Display label.
T3=tf(numt3,dent3)                   % Create and display T3.
[y3,t3]=step(T3);                    % Run step response of T3 and
                                     % collect points.
clf                                  % Clear graph.
plot(t1,y1,t2,y2,t3,y3)              % Plot acquired points with all
                                     % three plots on one graph.
title ('Step Responses of T1(s),T2 (s), and T3(s)')
                                     % Add title to graph.
xlabel('Time(seconds)')              % Add time axis label.
ylabel ('Normalized Response')       % Add response axis label.
text(0.7,0.7,'c3(t)')                % Label step response of T1.
text(0.7,1.1,'c2(t)')                % Label step response of T2.
text(0.5,1.3,'c1(t)')                % Label step response of T3.
pause
step(T1,T2,T3)                       % Use alternate method of plotting
                                     % step responses.
title ('Step Responses of T1(s), T2(s), and T3(s)')
                                     % Add title to graph.
pause
```

ch4p3 We also can plot the step response of systems represented in state space using the `step(T, t)` command. Here `T` is any LTI object and `t=a: b: c` is the range for the time axis, where `a` is the initial time, `b` is the time step size, and `c` is the final time. For example, `t=0:1:10` means time from 0 to 10 seconds in steps of 1 second. The `t` field is optional. Finally, in this example we introduce the command `grid on`, which superimposes a grid over the step response. Place the `grid on` command after the `step(T, t)` command.

```
'(ch4p3)'                            % Display label.
clf                                  % Clear graph.
A=[0 1 0; 0 0 1; -24 -26 -9];        % Generate A matrix.
B=[0; 0; 1];                         % Generate B vector.
C=[2 7 1];                           % Generate C vector.
D=0;                                 % Generate D.
T=ss(A,B,C,D)                        % Generate LTI object, T, in state
                                     % space and display.
t=0:0.1:10;                          % Define range of time for plot.
step(T,t)                            % Plot step response for given
                                     % range of time.
grid on                              % Turn grid on for plot.
pause
```

ch4p4 (Antenna Control Case Study) We now use MATLAB to plot the step response requested in the Antenna Control Case Study.

```
'(ch4p4) Antenna Control Case Study'
                                     % Display label.
clf                                  % Clear graph.
```

```
numg=2 0 .83;                          % Define numerator of G(s).
deng=[1 101.71 171];                   % Define denominator of G(s).
'G(s)'                                 % Display label.
G=tf (numg,deng)                       % Form and display transfer
                                       % Function G(s).
step(G);                               % Generate step response.
title('Angular Velocity Response')
                                       % Add title.
pause
```

ch4p5 (UFSS Case Study) As a final example, let us use MATLAB to do the UFSS Case Study in the text (*Johnson, 1980*). We introduce table lookup to find the rise time. Using the `interp1 (y, t, y1)` command, we set up a table of values of amplitude, `y`, and time, `t`, from the step response and look for the value of time for which the amplitude is `y` 1=0.1 and 0.9. We also generate time response data over a defined range of time using `t=a:b:c` followed by `[y, t]=step(G, t)`. Here `G` is an LTI transfer-function object and `t` is the range for the time axis, where `a` is the initial time, `b` is the time step size, and `c` is the final time; `y` is the output.

```
'(ch4p5) UFSS Case Study'              % Display label.
clf                                    % Clear graph.
'(a)'                                  % Display label.
numg=0.0169;                           % Define numerator of 2nd order
                                       % approximation of G(s).
deng=[1 0.226 0.0169];                 % Define 2nd order term of
                                       % denominator of G(s).
'G(s)'                                 % Display label.
G=tf(numg,deng).                       % Create and display G(s).
omegan=sqrt(deng(3))                   % Find natural frequency.
zeta=deng(2)/(2*omegan)                % Find damping ratio.
Ts=4/(zeta*omegan)                     % Find settling time.
Tp=pi/(omegan*sqrt(1-zeta^2))          % Find peak time.
pos=exp/(-zeta*pi/sqrt(1-zeta^2))*100
                                       % Find percent overshoot.
t=0:0.1:35;                            % Limit time to find rise time. t=0
                                       % to 35 in steps of 0.1.
[y,t]=step(G,t);                       % Generate and save points of step
                                       % response over defined range of t.
Tlow=interp1(y,t,0.1);                 % Search table for time when
                                       % y=0 . 1*finalvalue.
Thi=interp1(y,t,0.9);                  % Search table for
                                       % time=0.9*finalvalue.
Tr=Thi-Tlow                            % Calculate rise time.
'(b)'                                  % Display label.
numc=0.125*[1 0.435];                  % Define numerator of C(s).
denc=conv(poly([0-1.23]),[1 0.226 0.0169]);
                                       % Define denominator of C(s).
[K,p,k]=residue(numc, denc)            % Find partial-fraction expansion.
'(d)'                                  % Display label.
numg=0.125*[1 0.435];                  % Define numerator of G(s).
deng=conv([11.23],[1 0.226 0.0169]);
                                       % Define denominator of G(s).
'G(s)'                                 % Display label.
G=tf(numg,deng)                        % Create and display G(s).
```

```
[y,t]=step(G);                          % Generate complete step response
                                        % and collect points.
plot(t,y)                               % Plot points.
title('Pitch Angle Response')           % Add title.
xlabel('Time (seconds)')                % label time axis.
ylabel('Pitch Angle (radians)')         % Label y-axis.
pause
```

Chapter 5: Reduction of Multiple Subsystems

ch5p1 (UFSS Pitch Control System) MATLAB can be used for block diagram reduction. Three methods are available: (1) Solution via Series, Parallel, & Feedback Commands, (2) Solution via Algebraic Operations, and (3) Solution via Append & Connect Commands. Let us look at each of these methods.

1. **Solution via Series, Parallel, & Feedback Commands**
   The closed-loop transfer function is obtained using the following commands successively, where the arguments are LTI objects: `series (G1, G2)` for a cascade connection of $G_1(s)$; and $G_2(s)$; `parallel (G1, G2)` for a parallel connection of $G_1(s)$ and $G_2(s)$; `feedback(G,H, sign)` for a closed-loop connection with $G(s)$ as the forward path, $H(s)$ as the feedback, and `sign` is -1 for negative-feedback systems or $+1$ for positive-feedback systems. The sign is optional for negative-feedback systems.
2. **Solution via Algebraic Operations**
   Another approach is to use arithmetic operations successively on LTI transfer functions as follows: `G2*G1` for a cascade connection of $G_1(s)$ and $G_2(s)$; `G1+G2` for a parallel connection of $G_1(s)$ and $G_2(s)$; `G/(1+G*H)` for a closed-loop negative-feedback connection with $G(s)$ as the forward path and $H(s)$ as the feedback; `G/(1-G*H)` for positive-feedback systems. When using division we follow with the function `minreal (sys)` to cancel common terms in the numerator and denominator.
3. **Solution via Append & Connect Commands**
   The last method, which defines the topology of the system, may be used effectively for complicated systems. First, the subsystems are defined. Second, the subsystems are appended, or gathered, into a multiple-input/multiple-output system. Think of this system as a single system with an input for each of the subsystems and an output for each of the subsystems. Next, the external inputs and outputs are specified. Finally, the subsystems are interconnected. Let us elaborate on each of these steps.

   The subsystems are defined by creating LTI transfer functions for each. The subsystems are appended using the command `G=append (G1, G2, G3, G4, . . . . Gn)`, where the `Gi` are the LTI transfer functions of the subsystems and `G` is the appended system. Each subsystem is now identified by a number based upon its position in the append argument. For example, `G3` is 3, based on the fact that it is the third subsystem in the append argument (not the fact that we write it as `G3`).

   Now that we have created an appended system, we form the arguments required to interconnect their inputs and outputs to form our system. The first step identifies which subsystems have the external input signal and which subsystems have the external output signal. For example, we use `inputs=[1 5 6]` and `outputs=[3 4]` to define the external inputs to be the inputs of

subsystems 1,5, and 6 and the external outputs to be the outputs of subsystems 3 and 4. For single-input/single-output systems, these definitions use scalar quantities. Thus `inputs=5, outputs=8` define the input to subsystem 5 as the external input and the output of subsystem 8 as the external output.

At this point we tell the program how all of the subsystems are interconnected. We form a Q matrix that has a row for each subsystem whose input comes from another subsystem's output. The first column contains the subsystem's number. Subsequent columns contain the numbers of the subsystems from which the inputs come. Thus, a typical row might be as follows: `[3 6 -7]`, or subsystem 3's input is formed from the sum of the output of subsystem 6 and the negative of the output of subsystem 7.

Finally, all of the interconnection arguments are used in the `connect (G Q inputs, outputs)` command, where all of the arguments have been previously defined.

Let us demonstrate the three methods for finding the total transfer function by looking at the back endpapers and finding the closed-loop transfer function of the pitch control loop for the UFSS with $K_1 = K_2 = 1$ (*Johnson, 1980*). The last method using append and connect requires that all subsystems be proper (the order of the numerator cannot be greater than the order of the denominator). The pitch rate sensor violates this requirement. Thus, for the third method, we perform some block diagram maneuvers by pushing the pitch rate sensor to the left past the summing junction and combining the resulting blocks with the pitch gain and the elevator actuator. These changes are reflected in the program. You should verify all computer results with hand calculations.

```
'(ch5p1) UFSS Pitch Control System'
'& Feedback Commands'
'Solution via Series, Parallel'      % Display labels.
numg1=[-1];                          % Define numerator of G1(s).
deng1=[1];                           % Define denominator of G1(s).
numg2=[0 2];                         % Define numerator of G2(s).
deng2=[1 2];                         % Define denominator of G2 (s).
numg3=-0.125*[1 0.435];              % Define numerator of G3(s).
deng3=conv([1 1.23],[1 0.226 0.0169]);
                                     % Define denominator of G3(s).
numh1=[-1 0];                        % Define numerator of H1(s).
denh1=[0 1];                         % Define denominator of H1(s).
G1=tf(numg1,deng1);                  % Create LTI transfer function,
                                     % G1(s).
G2=tf(numg2,deng2);                  % Create LTI transfer function,
                                     % G2(s).
G3=tf(numg3,deng3);                  % Create LTI transfer function,
                                     % G3(s).
H1=tf (numh1,denh1);                 % Create LTI transfer function,
                                     % H1(s).
G4=series(G2,G3);                    % Calculate product of elevator
                                     % and vehicle dynamics.
G5=feedback(G4,H1);                  % Calculate close-loop transfer
                                     % function of inner loop.
Ge=series(G1,G5);                    % Multiply inner-loop transfer
                                     % function and pitch gain.
```

```
'T(s) via Series, Parallel, & Feedback Commands'
                                    % Display label.
T=feedback(Ge, 1)                   % Find closed-loop transfer
                                    % function.
' Solution via Algebraic Operations'
                                    % Display label.
clear                               % Clear session.
numg1=[-1];                         % Define numerator of G1(s).
deng1=[1];                          % Define denominator of G1(s).
numg2=[0 2];                        % Define numerator of G2(s).
deng2=[1 2];                        % Define denominator of G2(s).
numg3=-0.125*[1 0.435];             % Define numerator of G3(s).
deng3=conv([1 1.23],[1 0.226 0.0169]);
                                    % Define denominator of G3(s).
numh1=[-10];                        % Define numerator of H1(s).
denh1=[01];                         % Define denominator of H1(s).
G1=tf(numg1, deng1);                % Create LTI transfer function, G1(s).
G2=tf (numg2,deng2);                % Create LTI transfer function, G2(s).
G3=tf (numg3,deng3);                % Create LTI transfer function, G3(s).
H1=tf (numh1,denh1);                % Create LTI transfer function, H1(s).
G4=G3*G2;                           % Calculate product of elevator and
                                    % vehicle
                                    % dynamics.
G5=G4/(1+G4*H1);                    % Calculate closed-loop transfer
                                    % function of inner loop.
G5=minreal(G5);                     % Cancel common terms.
Ge=G5*G1;                           % Multiply inner-loop transfer
                                    % functions.
' T(s) via Algebraic Operations'    % Display label.
T=Ge/(1+Ge);                        % Find closed-loop transfer function.
T=minreal(T)                        % Cancel common terms.
' Solution via Append & Connect Commands'
                                    % Display label.
'G1(s)=(-K1)*(1/(-K2s))=1/s'        % Display label.
numg1=[1];                          % Define numerator of G1(s).
deng1=[1 0];                        % Define denominator of G1(s).
G1=tf(numg1,deng1)                  % Create LTI transfer function,
                                    % G1(s)=pitch gain*
                                    % 1 (1/Pitch rate sensor).
'G2(s)=(-K2s)*(2/(s+2)'             % Display label.
numg2=[-2 0];                       % Define numerator of G2(s).
deng2=[1 2];                        % Define denominator of G2(s).
G2=tf(numg2,deng2)                  % Create LTI transfer function,
                                    % G2(s)=pitch rate sensor*vehicle
                                    % dynamics.
'G3(s)=-0.125(s +0.435)/((s+1.23)(s^2+0.226s+0.0169))'
                                    % Display label.
numg3=-0.125*[1 0.435];             % Define numerator of G3(s).
deng3=conv([1 1.23],[1 0.226 0.0169]);
                                    % Define denominator of G3(s).
G3=tf(numg3,deng3);                 % Create LTI transfer function,
                                    % G3(s)=vehicle dynamics.
System=append(G1,G2,G3);            % Gather all subsystems.
input=1;                            % Input is at first subsystem,
```

```
                                        % G1(s).
output=3;                               % Output is output of third
                                        % subsystem, G3(s).
Q=[1 -3 0                               % Subsystem 1, G1(s) , gets its
                                        % input from the negative of the
                                        % output of subsystem 3, G3(s).
2 1-3                                   % Subsystem 2 , G2(s) , gets its
                                        % input from subsystem 1, G1(s) ,
                                        % and the negative of the output
                                        % of subsystem 3 , G3(s).
3 2 0];                                 % Subsystem 3 , G3(s) , gets its
                                        % input from subsystem 2 , G2(s).
T=connect(System, Q, input, output);
                                        % Connect the subsystems.
'T(s) via Append & Connect Commands'
                                        % Display label.
T=tf (T);                               % Create LTI closed-loop transfer
                                        % function.
T=minreal(T)                            % Cancel common terms.
pause
```

ch5p2 (Example 5.3) We can use MATLAB to calculate the closed-loop characteristics of a second-order system, such as damping ratio, ζ; natural frequency, ω_n; percent overshoot, $\%OS$ (pos); settling time, T_s; and peak time, T_p. The command `[numt, dent]=tfdata(T,'v')` extracts the numerator and denominator of $T(s)$ for a single-input/single-output system from which the calculations are based. The argument `'v'` returns the numerator and denominator as simple row vectors. Omitting `'v'` would return the numerator and denominator as cell arrays requiring more steps to obtain the row vectors. We end by generating a plot of the closed-loop step response. Let us look at Example 5.3 in the text.

```
'(ch5p2) Example 5.3'                   % Display label.
numg=[25];                              % Define numerator of G(s).
deng=poly([0 -5]);                      % Define denominator of G(s).
'G(s)'                                  % Display label.
G=tf(numg,deng).                        % Create and display G (s).
'T(s)'                                  % Display label.
T=feedback(G,1)                         % Find T(s).
[numt,dent]=tfdata(T,'v');              % Extract numerator & denominator
                                        % of T(s).
wn=sqrt(dent(3))                        % Find natural frequency.
z=dent(2)/(2*wn)                        % Find damping ratio.
Ts=4/(z*wn)                             % Find settling time.
Tp=pi/(wn*sqrt(1-z^2))                  % Find peak time.
pos=exp(-z*pi/sqrt(1-z^2))*100          % Find percent
                                        overshoot.
step(T)                                 % Generate step response.
pause
```

ch5p3 MATLAB can be used to convert transfer functions to state space in a specified form. The command `[Acc Bcc Ccc Dcc]=tf2ss (num, den)` can be used to convert $T(s)$=`num/den` into controller canonical form with matrices and vectors `Acc`, `Bcc`, `Ccc`, and `Dcc`. We can then form an LTI state-space object using `Scc=ss (Acc, Bcc, Ccc, Dcc)`. This object can then be converted into parallel

form using `Sp=canon (Scc, 'type')`, where `type=modal` yields the parallel form. Another choice, not used here, is `type=companion`, which yields a right companion system matrix. Transformation matrices can be used to convert to other representations. As an example, let us convert $C(s)/R(s) = 24/[(s+2)(s+3)(s+4)]$ into a parallel representation in state space, as is done in Section 5.7 - Parallel Form. Notice that the product of values in the **B** and **C** vectors yields the same product as the results in Eqs. (5.49) and (5.50). Thus, the two solutions are the same, but the state variables are ordered differently, and the gains are split between the **B** and **C** vectors. We can also extract the system matrices from the LTI object using `[A,B,C, D]=ssdata(S)`, where `S` is a state-space LTI object and `A, B, C, D,` are its associated matrices and vectors.

```
'(ch5p3)'                              % Display label.
numt=24;                               % Define numerator of T(s).
dent=poly([-2 -3 -4]);                 % Define denominator of T(s).
'T(s)'                                 % Display label.
T=tf(numt, dent)                       % Create and display T(s).
[Acc Bcc Ccc Dcc]=tf2ss (numt,dent);
                                       % Convert T(s) to controller
                                       % canonical form.
Scc=ss (Acc,Bcc,Ccc,Dcc);              % Create LTI controller canonical
                                       % state-space object.
Sp=canon(Scc,'modal');                 % Convert controller canonical form
                                       % to parallel form.
'Controller Canonical Form'            % Display label.
[Acc,Bcc,Ccc,Dcc]=ssdata(Scc)          % Extract and display controller
                                       % canonical form matrices.
'Parallel Form'                        % Display label.
[Ap,Bp,Cp,Dp]=ssdata(Sp)               % Extract and display parallel form
                                       % matrices.
pause
```

ch5p4 (Example 5.9) We can use MATLAB to perform similarity transformations to obtain other forms. Let us look at Example 5.9 in the text.

```
' (ch5p4) Example 5.9'                 % Display label.
Pinv=[2 0 0; 3 2 0; 1 4 5];            % Define P inverse.
P=inv(Pinv)                            % Calculate P.
'Original'                             % Display label.
Ax=[0 1 0;0 0 1;-2 -5 -7]              % Define original A.
Bx=[0 0 1]                             % Define original B.
Cx=[1 0 0]                             % Define original C.
'Transformed'                          % Display label.
Az=Pinv*Ax*P                           % Calculate new A.
Bz=Pinv*Bx                             % Calculate new B.
Cz=Cx*P                                % Calculate new C.
pause
```

ch5p5 Using MATLAB's `[P,d]=eig(A)` command, where the columns of `P` are the eigenvectors of `A` and the diagonal elements of `d` are the eigenvalues of `A`, we can find the eigenvectors of the system matrix and then proceed to diagonalize the system. We can also use `canon (S, 'modal')` to diagonalize an LTI object, `S`, represented in state space.

```
'(ch5p5)'                          % Display label.
A=[3 1 5;4 -2 7;2 3 1];            % Define original A.
B=[1;2;3];                         % Define original B.
C=                                 [2 4 6];
% Define original C.
[P,d]=eig(A)                       % Generate transformation matrix,
                                   % P, and eigenvalues, d.
'Via Transformation'               % Display label.
Adt=inv(P)*A*P                     % Calculate diagonal system A.
Bdt=inv(P)*B                       % Calculate diagonal system B.
Cdt=C*P                            % Calculate diagonal system C.
'Via Canon Command'                % Display label.
S=ss (A,B,C,0)                     % Create state-space LTI object
                                   % for original system.
Sp=canon(S,'modal')                % Calculate diagonal system via
                                   % canon command.
pause
```

Chapter 6: Stability

ch6p1 (Example 6.7) MATLAB can solve for the poles of a transfer function in order to determine stability. To solve for the poles of $T(s)$ use the `pole (T)` command. Let us look at Example 6.7 in the text.

```
'(ch6p1) Example 6.7'              % Display label.
numg=1;                            % Define numerator of G(s).
deng=conv ([1 0],[2 3 2 3 2]);     % Define denominator of G (s).
G=tf(numg,deng);                   % Create G(s) object.
'T(s)'                             % Display label.
T=feedback(G, 1)                   % Calculate closed-loop T(s)
                                   % object.
                                   % Negative feedback is default
                                   % when there is no sign parameter.
poles=pole(T)                      % Find poles of T(s).
pause
```

ch6p2 (Example 6.9) We can use MATLAB to find the range of gain for stability by generating a loop, changing gain, and finding at what gain we obtain right-half-plane poles.

```
'(ch6p2) Example 6.9'              % Display label.
K=[1:1:2000];                      % Define range of K from 1 to 2000
                                   % in steps of 1.
for n=1:1:length(K);               % Set up length of DO LOOP to equal
                                   % number of K values to be tested.
          dent=[1 18 77 K(n)];     % Define the denominator of T(s)
                                   % for the nth value of K.
          poles=roots(dent);       % Find the poles for the nth value
                                   % of K.
          r=real(poles);           % Form a vector containing the real
                                   % parts of the poles for K(n).
               if max(r) >=0,      % Test poles found for the nth
                                   % value of K for a real value ≥ 0.
                 poles             % Display first pole values where
                                   % there is a real part ≥ 0.
```

```
                K=K(n)              % Display corresponding value of K.
                break               % Stop loop if rhp poles are found.
          end                       % End if.
end                                 % End for.
pause
```

ch6p3 (Example 6.11) We can use MATLAB to determine the stability of a system represented in state space by using the command `eig (A)` to find the eigenvalues of the system matrix, `A`. Let us apply the concept to Example 6.11 in the text.

```
'(ch6p3) Example 6.11'              % Display label.
A=[0 3 1;2 8 1;-10 -5 -2]           % Define system matrix, A.
eigenvalues=eig(A)                  % Find eigenvalues.
pause
```

Chapter 7: Steady-State Errors

ch7p1 (Example 7.4, sys. b) Static error constants are found using $\lim s^n G(s)$ as $s \to 0$. Once the static error constant is found, we can evaluate the steady-state error. To evaluate the static error constant we can use the command `dcgain (G)`, which evaluates $G(s)$ at $s = 0$. Let us look at Example 7.4, system (b), in the text.

```
'(ch7p1) Example 7.4, sys . b'      % Display label
numg=500*poly([-2 -5 -6]);          % Define numerator of G(s).
deng=poly([0 -8 -10 -12]);          % Define denominator of G(s).
G=tf(numg,deng);                    % Form G(s)
'Check Stability'                   % Display label.
T=feedback(G,1);                    % Form T(s)
poles=pole(T)                       % Display closed-loop poles.
'Step Input'                        % Display label.
Kp=dcgain(G)                        % Evaluate Kp=numg/deng for s=0.
ess=1/(1+Kp)                        % Evaluate ess for step input.
'Ramp Input'                        % Display label.
numsg=conv([1 0],numg);             % Define numerator of sG(s).
densg=poly([0 -8 -10 -12]);         % Define denominator of sG(s).
sG=tf(numsg,densg);                 % Create sG(s).
sG=minreal(sG);                     % Cancel common 's' in
                                    % numerator(numsg) and
                                    % denominator(densg).
Kv=dcgain(sG)                       % Evaluate Kv=sG(s) for s=0.
ess=1/Kv                            % Evaluate steady-state error for
                                    % ramp input.
'Parabolic Input'                   % Display label.
nums2g=conv([1 0 0],numg);          % Define numerator of s^2G(s).
dens2g=poly([0 -8 -10-12]);         % Define denominator of s^2G(s).
s2G=tf(nums2g,dens2g);              % Create s^2G(s).
s2G=minreal(s2G);                   % Cancel common 's' in
                                    % numerator(nums2g) and
                                    % denominator(dens2g).
Ka=dcgain(s2G)                      % Evaluate Ka=s^2G(s) fors=0.
ess=1/Ka                            % Evaluate steady-state error for
                                    % parabolic input.
pause
```

ch7p2 (Example 7.6) We can use MATLAB to evaluate the gain, K, required to meet a steady-state error specification. Let us look at Example 7.6 in the text.

```
'(ch7p2) Example 7.6'                % Display label.
numgdK=[1 5];                        % Define numerator of G(s)/K.
dengdK=poly ([0 -6 -7 -8]);          % Define denominator of G(s)/K.
GdK=tf(numgdK,dengdK);               % Create G(s)/K.
numgkv=conv ([1 0], numgdK);         % Define numerator of sG(s)/K.
dengkv=dengdK;                       % Define denominator of sG(s)/K.
GKv=tf(numgkv,dengkv);               % Create sG(s)/K.
GKv=minreal(GKv);                    % Cancel common 's' in numerator
                                     % and denominator of sG(s)/K.
KvdK=dcgain(GKv)                     % Evaluate (Kv/K)=(numgkv/dengkv)
                                     % for s=0.
ess=0.1                              % Enumerate steady-state error.
K=1/(ess*KvdK)                       % Solve for K.
'Check Stability'                    % Display label.
T=feedback(K*GdK,1);                 % Form T(s).
poles=pole(T)                        % Display closed-loop poles.
pause
```

Chapter 8: Root Locus Techniques

ch8p1 (Example 8.7) MATLAB allows root loci to be plotted with the `rlocus (GH)` command, where $G(s)H(s)$=`numgh/dengh` and GH is an LTI transfer-function object. Points on the root locus can be selected interactively using the `[K,p] =rlocfind (GH)` command. MATLAB then yields the gain `(K)` at that point as well as all other poles (`p`) that have that gain. We can zoom in and out of the root locus by changing the range of axis values using the command `axis ([xmin, xmax, ymin, ymax])`. The root locus can be drawn over a grid that shows constant damping ratio `(z)` and constant natural frequency `(wn)` curves using the `sgrid (z, wn)` command. To plot multiple ζ and ω_n curves, use `z=zmin:zstep:zmax` and `wn=wnmin:wn-step:wnmax` to specify ranges of values.

```
'(ch8p1) Example 8.7'                % Display label.
clf                                  % Clear graph on screen.
numgh=[1-4 20];                      % Define numerator of G (s) H (s).
dengh=poly([-2 -4]);                 % Define denominator of G (s) H (s).
'G(s)H(s)'                           % Display label.
GH=tf(numgh,dengh)                   % Create G(s)H(s) and display.
rlocus(GH)                           % Draw root locus.
z=0.2:0.05:0.5;                      % Define damping ratio values : 0.2
                                     % to 0.5 in steps of 0.05.
wn=0:1:10;                           % Define natural frequency values:
                                     % 0 to 10 in steps of 1.
sgrid(z,wn)                          % Generate damping ratio and
                                     % natural frequency grid lines for
                                     % root locus.
title ('Root Locus')                 % Define title for root locus.
pause
rlocus(GH)                           % Draw close-up root locus.
axis([-3 1-4 4])                     % Define range on axes for root
                                     % locus close-up view.
title('Close-up')                    % Define title for close-up root
                                     % locus.
```

```
z=0.45;                              % Define damping ratio line for
                                     % overlay on close-up root locus.
wn=0;                                % Suppress natural frequency
                                     % overlay curves.
sgrid(z,wn)                          % Overlay damping ratio curve on
                                     % close-up root locus.
for k=1: 3                           % Loop allows 3 points to be
                                     % selected as per Example 8.7,
                                     %(z=0.45 , jwcrossing,breakaway).
[K,p]=rlocfind(GH)                   % Generate gain, K, and closed-loop
                                     % poles, p, for point selected
                                     % interactively on the root locus.
end                                  % End loop.
pause
```

ch8p2 (Example 8.8) We can couple the design of gain on the root locus with a step-response simulation for the gain selected. We introduce the command `rlocus (G, K)`, which allows us to specify the range of gain, `K`, for plotting the root locus. This command will help us smooth the usual root locus plot by equivalently specifying more points via the argument, `K`. Notice that the first root locus plotted without the argument `K` is not smooth. We also introduce the command `x=input('prompt')`, which allows keyboard entry of a value for `x` in response to a prompt. We apply this command to enter the desired percent overshoot. We also add a variable's value to the title of the root locus and step-response plots by inserting another field in the title command and use `num2str (value)` to convert value from a number to a character string for display. Let us apply the concepts to Example 8.8 in the text.

```
'(ch8p2) Example 8.8'                % Display label.
clear                                % Clear variables from workspace.
clf                                  % Clear graph on screen.
numg=[11.5];                         % Define numerator of G (s).
deng=poly ([0 -1 -10]);              % Define denominator of G(s).
'G(s)'                               % Display label.
G=tf (numg, deng)                    % Create and display G (s).
rlocus (G)                           % Draw root locus (H(s)=1).
title('Original Root Locus')         % Add title.
pause
K=0.005;                             % Specify range of gain to smooth
                                     % root locus,
rlocus (G,K)                         % Draw smoothed root locus
                                     % (H(s)=1).
title (' Smoothed Root Locus')       % Add title.
pos=input('Type %OS');               % Input desired percent overshoot
                                     % from the keyboard.
z=-log(pos/100)/sqrt(pi^2+[log(pos/100)]^2)
                                     % Calculate damping ratio.
sgrid(z , 0)                         % Overlay desired damping ratio
                                     % line on root locus.
title(['Root Locus with', num2str (pos) ,' % overshoot line'])
                                     % Define title for root locus
                                     % showing percent overshoot used.
[K, p]=rlocfind(G)                   % Generate gain, K, and closed-
                                     % loop poles, p, for point selected
                                     % interactively on the root locus.
```

```
pause
'T(s)'                                  % Display label
T=feedback(K*G, 1)                      % Find closed-loop transfer
                                        % function
                                        % with selected K and display.
step(T)                                 % Generate closed-loop step
                                        % response for point select on
                                        % root locus.
title(['Step Response for K=',num2str(K)])
                                        % Give step response a title which
                                        % includes the value of K.
pause
```

Chapter 9:Design Via Root Locus

ch9p1 (Example 9.3) We can use MATLAB to design PD controllers. The program allows us to input a desired percent overshoot via the keyboard. MATLAB then produces a root locus for the uncompensated system with an overlay of the percent overshoot line. We interactively select the intersection of the root locus and the desired percent overshoot line to set the gain. MATLAB outputs an estimate of the uncompensated system's performance specifications and a step response of the uncompensated system for us to determine the required settling time. After we input the settling time through the keyboard, MATLAB designs the PD controller and produces a root locus of the PD compensated system from which we can interactively select the gain. Finally, MATLAB produces an estimate of the PD compensated system's performance specifications and a step response of the PD compensated system.

```
'(ch9p1) Example 9.3'                   % Display label.
clf                                     % Clear graph on screen.
'Uncompensated System'                  % Display label.
numg=1;                                 % Generate numerator of G(s).
deng=poly([0 -4 -6]);                   % Generate denominator of G(s).
'G(s)'                                  % Display label.
G=tf(numg, deng)                        % Create and display G(s).
pos=input('Type desired percent overshoot');
                                        % Input desired percent overshoot.
z-log(pos/100)/sqrt(pi^2+[log(pos/100)]^2);
                                        % Calculate damping ratio.
rlocus(G)                               % Plot uncompensated root locus.
sgrid(z, 0)                             % Overlay desired percent
                                        % overshoot line.
title(['Uncompensated Root Locus with',num2str(pos),...
Overshoot Line'])                       % Title uncompensated root locus.
[K,p]=rlocfind(G);                      % Generate gain, K, and closed-loop
                                        % poles, p, for point selected
                                        % interactively on the root locus.
'Closed-loop poles='                    % Display label.
p                                       % Display closed-loop poles.
f=input('Give pole number that is operating point');
                                        % Choose uncompensated system
                                        % dominant pole.
'Summary of estimated specifications for selected point on'
'uncompensated root locus'              % Display label.
operatingpoint=p(f)                     % Display uncompensated dominant
                                        % pole.
```

```
gain=K                              % Display uncompensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                    % Display uncompensated settling
                                    % time.
estimated_peak_time=pi/abs(imag(p(f)))
                                    % Display uncompensated peak time.
estimated_percent_overshoot=pos
                                    % Display uncompensated percent
                                    % overshoot.
estimated_damping_ratio=z           % Display uncompensated damping
                                    % ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2)
                                    % Display uncompensated natural
                                    % frequency.
numkv=conv([1 0],numg);             % Set up numerator to evaluate Kv.
denkv=deng;                         % Set up denominator to evaluate Kv.
sG=tf(numkv,denkv);                 % Create sG(s).
sG=minreal (sG);                    % Cancel common poles and zeros.
Kv=dcgain(K*sG)                     % Display uncompensated Kv.
ess=1/Kv                            % Display uncompensated
                                    % steady-state
                                    % error for unit ramp input.
'T(s)'                              % Display label.
T=feedback(K*G, 1)                  % Find uncompensated T(s).
step(T)                             % Plot step response of
                                    % uncompensated system.
title (['Uncompensated System Step Response with',num2str (pos) , . . .
'% Overshoot'])                     % Add title to uncompensated step
                                    % response.
'Press any key to go to PD compensation'
                                    % Display label.
pause
'Compensated system'                % Display label.
Ts=input('Type Desired Settling Time');
                                    % Input desired settling time from
                                    % the keyboard.
wn=4/(Ts*z);                        % Calculate natural frequency.
desired_pole=(-z*wn)+(wn*sqrt(1-z^2)*i);
                                    % Calculate desired dominant pole
                                    % location.
angle_at_desired_pole=(180/pi)*...
angle(polyval(numg, desired_pole)/polyval(deng,desired_pole));
                                    % Calculate angular contribution
                                    % to desired pole without PD
                                    % compensator.
PD_angle=180-angle_at_desired_pole;
                                    % Calculate required angular
                                    % contribution from PD
                                    % compensator.
zc=((imag(desired_pole)/tan(PD_angle*pi/180))...
_real (desired_pole));              % Calculate PD zero location.
'PD' Compensator'                   % Display label.
numc=[1 zc];                        % Calculate numerator of Gc (s).
denc=[0 1];                         % Calculate numerator of Gc (s).
'Gc(s)'                             % Display label.
```

```
Gc=tf (numc, denc)                    % Create and display Gc (s).
'G(s)Gc(s)'                           % Display label.
Ge=G*Gc                               % Cascade G(s) and Gc (s).
rlocus (Ge,0:0.005:100)               % Plot root locus of PD compensated
                                      % system.
syrid(z,0)                            % Overlay desired percent
                                      % overshoot line.
title (['PD Compensated Root Locus with', num2str (pos) ,...
' % Overshoot Line'])                 % Add title to PD compensated root
                                      % locus.
[K,p]=rlocfind(Ge);                   % Generate gain, K, and closed-loop
                                      % poles, p, for point selected
                                      % interactively on the root locus.
'Closed-loop poles='                  % Display label.
p                                     % Display PD compensated system's
                                      % closed-loop poles.
f=input('Give pole number that is operating point');
                                      % Choose PD compensated system
                                      % dominant pole.
'Summary of estimated specifications for selected point on PD'
'compensated root locus'              % Display label.
operatingpoint=p(f)                   % Display PD compensated dominant
                                      % pole.
gain=K                                % Display PD compensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                      % Display PD compensated settling
                                      % time.
estimated_peak_time=pi/abs(imag(p(f)))
                                      % Display PD compensated peak time.
estimated_percent_overshoot=pos       % Display PD compensated percent
                                      % overshoot.
estimated_damping_ratio=z             % Display PD compensated damping
                                      % ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2
                                      % Display PD compensated natural
                                      % frequency.
s=tf ([1 0], 1);                      % Created transfer function,'s' .
sGe=s*Ge;                             % Create sGe(s).
sGe=minreal (sGe);                    % Cancel common poles and zeros.
Kv=dcgain(K*sGe)                      % Display compensated Kv.
ess=1/Kv                              % Display compensated
                                      % steady-state error for
                                      % unit ramp input.
'T(s)'                                % Display label.
T=feedback(K*Ge, 1)                   % Create and display PD compensated
%T(s).
'Press any key to continue and obtain the PD compensated step'
'response'                            % Display label.
pause
step(T)                               % Plot step response for PD
                                      % compensated system.
title ([' PD Compensated System Step Response with'...
num2str (pos) ,'% Overshoot'])        % Add title to step response
                                      % of PD compensated system.
pause
```

ch9p2 (Example 9.4) We can use MATLAB to design a lead compensator. The program allows us to input a desired percent overshoot via the keyboard. MATLAB then produces a root locus for the uncompensated system with an overlay of the percent overshoot line. We interactively select the intersection of the root locus and the desired percent overshoot line to set the gain. MATLAB outputs an estimate of the uncompensated system's performance specifications and a step response of the uncompensated system for us to determine the required settling time. Next we input the settling time and the lead compensator zero through the keyboard. At this point we take a different approach from that of the previous example. Rather than letting MATLAB calculate the lead compensator pole directly, MATLAB produces a root locus for every interactive guess of a lead compensator pole. Each root locus contains the desired damping ratio and natural frequency curves. When our guess is correct, the root locus, the damping ratio line, and the natural frequency curve will intersect. We then interactively select this point of intersection to input the gain. Finally, MATLAB produces an estimate of the lead-compensated system's performance specifications and a step response of the lead-compensated system.

```
'(ch9p2) Example 9.4'                % Display label.
Clf                                  % Clear graph on screen.
'Uncompensated System'               % Display label.
numg=1;                              % Generate numerator of G (s).
deng=poly ([0 -4 -6]);               % Generate denominator of G(s).
'G(s)'                               % Display label.
G=tf (numg, deng)                    % Create and display G (s).
pos=input ('Type desired percent overshoot');
                                     % Input desired percent overshoot.
z=-log(pos/100)/sqrt(pi^2+[log(pos/100)]^2);
                                     % Calculate damping ratio.
rlocus (G)                           % Plot uncompensated root locus.
sgrid(z , 0)                         % Overlay desired percent
                                     % overshoot line.
title (['Uncompensated Root Locus with', num2str (pos),...
' % Overshoot Line'])                % Title uncompensated root locus.
[K,p]=rlocfind(G);                   % Generate gain, K, and closed-loop
                                     % poles, p, for point selected
                                     % interactively on the root locus.
'Closed-loop poles='                 % Display label.
p                                    % Display closed-loop poles.
f=input ('Give pole number that is operating point');
                                     % Choose uncompensated system
                                     % dominant pole.
'Summary of estimated specifications for selected point on'
'uncompensated root locus'           % Display label.
operatingpoint=p(f)                  % Display uncompensated dominant
                                     % pole.
gain=K                               % Display uncompensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                     % Display uncompensated settling
                                     % time.
estimated_peak_time=pi/abs(imag(p(f)))
                                     % Display uncompensated peak time.
estimated_percent_overshoot=pos      % Display uncompensated percent
                                     % overshoot.
```

```
estimated_damping_ratio=z          % Display uncompensated damping
                                   % ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2
                                   % Display uncompensated natural
                                   % frequency.
numkv=conv([1 0],numg);            % Set up numerator to evaluate Kv.
denkv=deng;                        % Set up denominator to evaluate Kv.
sG=tf(numkv,denkv);                % Create sG(s).
sG=minreal (sG);                   % Cancel common poles and zeros.
Kv=dcgain(K*sG)                    % Display uncompensated Kv.
ess=1/Kv                           % Display uncompensated
                                   % steady-state error for
                                   % unit ramp input.
'T(s)'                             % Display label.
T=feedback(K*G, 1)                 % Create and display T(s).
step(T)                            % Plot step response of
                                   % uncompensated system.
title ([['Uncompensated System Step Response with',...
num2str(pos),'% Overshoot'])       % Add title to uncompensated step
                                   % response.
'Press any key to go to lead compensation'
                                   % Display label,
pause
Ts=input ('Type Desired Settling Time');
                                   % Input desired settling time.
b=input ('Type Lead Compensator Zero, (s+b). b= ');
                                   % Input lead compensator zero.
done=1;                            % Set loop flag.
while done==1                      % Start loop for trying lead
                                   % compensator pole.
a=input ('Enter a Test Lead Compensator Pole, (s+a). a= ');
                                   % Enter test lead compensator pole.
numge=conv (numg,[1 b]);           % Generate numerator of Gc(s)G(s).
denge=conv([1a],deng);             % Generate denominator
                                   % of Gc(s)G(s).
Ge=tf(numge,denge);                % Create Ge(s)=Gc(s)G(s).
wn=4/(Ts*z);                       % Evaluate desired natural
                                   % frequency.
clf                                % Clear graph on screen.
rlocus(Ge)                         % Plot compensated root locus with
                                   % test lead compensator pole.
axis([-10,10,-10,10])              % Change lead-compensated
                                   % root locus axes.
sgrid(z, wn)                       % Overlay grid on lead-compensated
                                   % root locus.
title (['Lead-Compensated Root Locus with', num2str (pos),...
                                   % Overshoot Line, Lead Pole at',...
num2str(-a),'and Required Wn'])    % Add title to lead-compensated
                                   % root locus.
done=input ('Are you done? (y=0,n=1)');
                                   % Set loop flag.
end                                % End loop for trying compensator
                                   % pole.
[K,p]=rlocfind (Ge);               % Generate gain, K, and closed-loop
                                   % poles, p, for point selected
```

```
                                    % interactively on the root locus.
'Gc(s)'                             % Display label.
Gc=tf ([1 b],[1 a])                 % Display lead compensator.
'Gc(s)G(s)'                         % Display label.
Ge                                  % Display Gc(s)G(s).
'Closed-loop poles='                % Display label.
p                                   % Display lead-compensated
                                    % system's
                                    % closed-loop poles.
f=input('Give pole number that is operating point');
                                    % Choose lead-compensated system
                                    % dominant pole.
'Summary of estimated specifications for selected point on lead'
'compensated root locus'            % Display label.
operatingpoint=p(f)                 % Display lead-compensated
                                    % dominant pole.
gain=K                              % Display lead-compensated gain.
estimated_settling_time=4/abs(real(p(f)))
                                    % Display lead-compensated
                                    % settling time.
estimated_peak_time=pi/abs(imag(p(f)))
                                    % Display lead-compensated
                                    % peak time.
estimated_percent_overshoot=pos     % Display lead-compensated
                                    % percent overshoot.
estimated_damping_ratio=z           % Display lead-compensated
                                    % damping ratio.
estimated_natural_frequency=sqrt(real(p(f))^2+imag(p(f))^2)
                                    % Display lead-compensated
                                    % natural frequency.
s=tf ([1 0], 1);                    % Create transfer Function,'s' .
sGe=s*Ge;                           % Create sGe (s) to evaluate Kv.
sGe=minreal (sGe);                  % Cancel common poles and zeros.
Kv=dcgain(K*sGe)                    % Display lead-compensated Kv.
ess=1/Kv                            % Display lead-compensated steady-
                                    % state error for unit ramp input.
'T (s)'                             % Display label.
t=feedback(K*Ge, 1)                 % Create and display lead-
                                    % compensated T (s).
'Press any key to continue and obtain the lead-compensated step'
'response'                          % Display label.
pause
step(T)                             % Plot step response for lead
                                    % compensated system.
title (['Lead-Compensated System Step Response with',...
num2str (pos),'% Overshoot'])       % Add title to step response
                                    % of lead-compensated system.
pause
```

Chapter 10: Frequency Response Techniques

ch10p1 (Example 10.3) We can use MATLAB to make Bode plots using `bode(G)`, where `G/(s)=numg/deng` and `G` is an LTI transfer-function object. Information about the plots obtained with `bode(G)` can be found by left-clicking the mouse on the curve. You can find the curve's label, as well as the coordinates of the point on which you clicked. Right-clicking away from a curve brings up a menu if the icons on the menu

bar are deselected. From this menu you can select (1) system responses to be displayed and (2) characteristics, such as peak response. When selected, a dot appears on the curve at the appropriate point. Let your mouse rest on the point to read the value of the characteristic. You may also select (3) which curves to view, (4) choice for grid on or off, (5) returning to full view after zooming, and (6) properties, such as labels, limits, units, style, and characteristics. We can obtain points on the plot using `[mag, phase, w]=bode(G)`, where magnitude, phase, and frequency are stored in `mag, phase, and w`, respectively. Magnitude and phase are stored as 3-D arrays. We use `mag(:,:)', phase(:,:)'` to convert the arrays to column vectors, where the apostrophe signifies matrix transpose. Let us look at Example 10.3 in the text.

```
'(ch10p1) Example 10.3'                % Display label.
clf                                    % Clear graph on screen.
numg=[1 3];                            % Define numerator of G(s).
deng=conv([1 2],[1 2 25]);             % Define denominator of G(s).
'G(s)'                                 % Display label.
G=tf(numg, deng)                       % Create and display G(s).
bode(G)                                % Make a Bode plot.
grid on                                % Turn on grid for Bode plot.
title('Open-Loop Frequency Response')
                                       % Add a title to the Bode plot.
[mag,phase,w]=bode(G);                 % Store points on the Bode plot.
points=[20*log10(mag(:,:))',phase(:,:)',w]
                                       % List points on Bode plot with
                                       % magnitude in dB.
pause
```

ch10p2 (Example 10.5) We can use MATLAB to make Nyquist diagrams using `nyquist(G)`, where $G(s)$=`numg/deng` and `G` is an LTI transfer-function object. Information about the plots obtained with `nyquist(G)` can be found by left-clicking the mouse on the curve. You can find the curveš label, as well as the coordinates of the point on which you clicked and the frequency. Right-clicking away from a curve brings up a menu if the icons on the menu bar are deselected. From this menu you can select (1) system responses to be displayed and (2) characteristics, such as peak response. When selected, a dot appears on the curve at the appropriate point. Let your mouse rest on the point to read the value of the characteristic. You may also select (3) whether or not to show negative frequencies, (4) choice for grid on or off, (5) choice for zooming to (–1,0), (6) returning to full view after zooming, and (7) properties, such as labels, limits, units, style, and characteristics. We can obtain points on the plot by using `[re, im, w]=nyquist(G)`, where the real part, imaginary part, and frequency are stored in `re, im, and w`, respectively, and `re` and im are 3-D arrays. We can specify a range of w by using `[re, im]=nyquist(G,w)`. We use `re(:,:)'`, and `im(:,:)'` to convert the arrays to column vectors. Let us look at Example 10.5 in the text.

```
'(ch10p2) Example 10.5'                % Display label.
clf                                    % Clear graph on screen.
numg=[1 2];                            % Define numerator of G(s).
deng=[1 0 0];                          % Define denominator of G(s).
'G(s)'                                 % Display label.
G=tf(numg, deng)                       % Create and display G(s).
nyquist(G)                             % Make a Nyquist diagram.
```

```
grid on                                % Turn on grid for Nyquist diagram.
title('Open-Loop Frequency Response')
                                       % Add a title to the Nyquist
                                       % diagram.
w=0:0.5:10;                            % Let 0 <w< 10 in steps of 0.5.
[re,im]=nyquist(G,w);                  % Get Nyquist diagram points for a
                                       % range of w.
points=[re(:,:)', im(:,:)',w']% List specified range of points
                                       % in Nyquist diagram.
pause
```

ch10p3 (Example 10.8) We can use MATLAB to find gain margin (`Gm`), phase margin (`Pm`), the gain margin frequency, where the phase plot goes through 180 degrees (`Wcg`), and the phase-margin frequency, where the magnitude plot goes through zero `dB (Wcp)`. To find these quantities we use `[`G_m`, `P_m`, `W_{cg}`, `W_{cp}`] =margin (G)`, where $G(s)$`=numg/deng` and `G` is an LTI transfer-function object. Let us look at Example 10.8 in the text.

```
'(ch10p3) Example 10.8'                % Display label.
clf                                    % Clear graph on screen.
numg=6;                                % Define numerator of G(s).
deng=conv ([1 2],[1 2 2]);             % Define denominator of G (s).
'G(s)'                                 % Display label.
G=tf (numg, deng)                      % Create and display G (s).
nyquist(G)                             % Make a Nyquist diagram.
grid on                                % Turn on grid for the Nyquist
                                       % diagram.
title ('Open-Loop Frequency Response')
                                       % Add a title to the Nyquist
                                       % diagram.
[Gm,Pm,Wcg,Wcp]=margin(G);             % Find margins and margin
                                       % frequencies.
'Gm(dB); Pm(deg.); 180 deg. freq.(r/s); 0 dB freq. (r/s)'
                                       % Display label.
margins=[20*log10(Gm),Pm,Wcg,Wcp]
                                       % Display margin data.
pause
```

ch10p4 (Example 10.9) We can use MATLAB to determine the range of K for stability using frequency response methods. Let us look at Example 10.9 in the text.

```
'(ch10p4) Example 10.9'                % Display label.
numg=1;                                % Define numerator of G(s).
deng=poly ([-2 -4 -5]);                % Define denominator of G(s).
'G(s)'                                 % Display label.
G=tf (numg, deng)                      % Create and display G(s).
[Gm,Pm,Wcg,Wcp]=margin(G);             % Find margins and margin
                                       % frequencies.
K=Gm                                   % Display K for stability.
pause
```

ch10p5 (Example 10.11) We can use MATLAB to find the closed-loop frequency response. Let us look at Example 10.11 in the text.

```
'(ch10p5) Example 10.11'          % Display label.
clf                               % Clear graph on screen.
numg=50;                          % Define numerator of G(s).
deng=poly ([0 -3 -6]);            % Define denominator of G(s).
'G(s)'                            % Display label.
G=tf (numg, deng)                 % Create and display G(s).
'T(s)'                            % Display label.
T=feedback(G, 1)                  % Find and display closed-loop
                                  % transfer function.
bode(T)                           % Make a Bode plot.
grid on                           % Turn on the grid for the plots.
title('Closed-Loop Frequency Response')
                                  % Add a title to the Bode plot,
pause
nyquist(T)                        % Make a Nyquist diagram.
title('Closed-Loop Frequency Response')
                                  % Add a title to the Nyquist
                                  % diagram.
pause
```

ch10p6 We can use MATLAB to plot Nichols charts using `nichols(G)`, where `G(s)=numg/deng` and `G` is an LTI transfer-function object. The Nichols grid can be added using the `ngrid` command after the `nichols(G)` command. Information about the plots obtained with `nichols(G)` can be found by left-clicking the mouse on the curve. You can find the curves' label, as well as the coordinates of the point on which you clicked and the frequency. Right-clicking away from a curve brings up a menu if the icons on the menu bar are deselected. From this menu you can select (1) system responses to be displayed and (2) characteristics, such as peak response. When selected, a dot appears on the curve at the appropriate point. Let your mouse rest on the point to read the value of the characteristic. You may also select (3) choice for grid on or off, (4) returning to full view after zooming, and (5) properties, such as labels, limits, units, style, and characteristics. Let us make a Nichols chart of $G(s) = 1/[s(s+1)(s+2)]$.

```
'(ch10p6)'                        % Display label.
clf                               % Clear graph on screen.
numg=1;                           % Define numerator of G(s).
deng=poly ([0 -1 -2]);            % Define denominator of G(s).
'G(s)'                            % Display label.
G=tf (numg, deng)                 % Create and display G(s).
nichols (G)                       % Make a Nichols plot.
ngrid                             % Add Nichols grid.
pause
```

ch10p7 (Example 10.15) We can use MATLAB and frequency response methods to include time delay in the loop. Time delay is represented by `[numd, dend]=pade (T, n)`, where `T` is the delay time in seconds and `n` is the order. Larger values of n give better approximations to the delay, G_d`(s)=numd/dend`. Since we are plotting multiple plots, we first collect the data for the Bode plots by using `[mag, phase] =bode (G, w)`, where `w` is specified as a range of frequencies. We then use the generic plotting command. Also notice the commands used to label the axes and the plots on the Bode plot (see the MATLAB instruction manual for details). Let us look at Example 10.15 in the text.

```
'(ch10p7) Example 10.15'            % Display label.
clf                                 % Clear graph on screen.
hold off                            % Turn graph hold off.
numg=1;                             % Define numerator of G(s).
deng=poly ([0 -1 -10]);             % Define denominator of G(s).
'G(s)'                              % Display label.
G=tf (numg, deng)                   % Create and display G (s).
w=0.01:0.1:10;                      % Let 0.01<w<10 in steps of 0.1.
[magg,phaseg]=bode (G,w);           % Collect Bode data for G(s).
[numd,dend]=pade(1,6);              % Represent the delay.
Gd=tf (numd,dend);                  % Create and display the delay,
                                    % Gd(s).
[magd,phased]=bode(Gd,w);           % Collect Bode data for Gd(s).
Ge=Gd*G;                            % Form Gd(s)G(s).
[mage,phasee]=bode(Ge,w);           % Collect Bode data for Gd(s)G(s).
subplot (2,1,1)                     % Subdivide plot area for plot 1.
semilogx(w,20*log10(mage(:,:)))     % Plot magnitude response.
grid on                             % Turn on grid for magnitude plot.
axis([0.01,10,-80,20]);             % Limit Bode plot axes.
title ('Magnitude Response with Delay')
                                    % Add title to magnitude response.
xlabel('Frequency (rad/s)')         % Label x-axis of magnitude
                                    % response.
ylabel('20log M')                   % Label y-axis of magnitude
                                    % response.
subplot (2,1,2)                     % Subdivide plot area for plot 2.
semilogx(w,phaseg(:,:),w,phased(:,:)-3*360,w,phasee(:,:)-3*360)
                                    % Plot phase response for G(s),
                                    % Gd (s), and G (s) Gd (s) on one
                                    % graph.
grid on                             % Turn on grid for phase plot.
axis([0.01,10,-900,0]);             % Limit Bode plot axes.
title ('Phase Response with Delay')
                                    % Add title to phase response.
xlabel('Frequency (rad/s)')         % Label x-axis of phase response.
ylabel('Phase (degrees)')           % Label y-axis of phase response.
text(1.5,-50,'Time Delay')          % Label time delay curve.
text(4,-150,'System')               % Label system curve.
text(2.7,-300,'Total')              % Label total curve.
pause
```

ch10p8 (Example 10.18) We can use MATLAB and frequency response methods to determine experimentally a transfer function from frequency response data. By determining simple component transfer functions and then successively subtracting their frequency response, we can approximate the complete transfer function. Let us look at Example 10.18 in the text and use MATLAB for a portion of the problem. You can complete the program for practice. For this problem we generate the original frequency response plot via a transfer function. Normally, the data for the original frequency response plot would be tabular, and the program would begin at the step `[M0, P0]=bode (G0, w)` where the tabular data is generated. In other words, in a real application, the data would consist of column vectors `M0, P0, and w`'.

```
'(ch10p8) Example 10.18'            % Display label.
clf                                 % Clear graph on screen.
```

```
hold off                                % Turn graph hold off.
% Generate the experimental Bode plots for G0 (s)=numg0/deng0, that
% is, M0, P0.
numg0=70*[1 20];                        % Define numerator of G0(s).
deng0=conv([1 7],[1 2 25]);             % Partially define denominator of
                                        % G0(s).
deng0=conv(deng0,[1 70]);               % Complete the denominator of
                                        % G0(s).
G0=tf(numg0,deng0);                     % Create G0(s).
w=1:0.5:1000;                           % Let 1<w<1000 in steps of 0.5.
[M0, P0]=bode(G0,w);                    % Generate the tabular data.
[20*log10(M0(:,:))',P0(:,:)',w'];
                                        % Convert magnitude data to dB.
bode(G0,w)                              % Generate a Bode plot.
grid on                                 % Turn on grid for Bode plot.
title('Experimental')                   % Add title.
pause
clf                                     % Clear graph.
% Estimate a component part of the transfer function as
% G1 (s)=25/ (s^2+2*0. 22*5s+5^2) and subtract it from the experimental
% frequency response
numg1=5^2;                              % Define numerator of G1(s).
deng1=[1 2*0.22*5 5^2];                 % Define denominator of G1(s).
'First estimate'                        % Display label.
G1=tf (numg1, deng1)                    % Create and display G1 (s).
[M1,P1]=bode(G1,w);                     % Generate Bode data for G1 (s).
M2=20*log10(M0(:,:))-20*log10(M1(:,:));
                                        % Subtract Bode magnitude data of
                                        % G1 from original magnitude data.
P2=P0(:,:)-P1(:,:);                     % Subtract Bode phase data of G1
                                        % from original phase data.
subplot(2,1,1)                          % Divide plot area in two for
                                        % magnitude plot.
semilogx(w(:,:),M2)                     % Plot magnitude response after
                                        % subtracting.
grid on                                 % Turn on grid for magnitude plot.
xlabel('Frequency (rad/sec)')           % Add x-axis label.
ylabel('Gain dB')                       % Add y-axis label.
subplot(2,1,2)                          % Divide plot area in two for phase
                                        % plot.
semilogx(w,P2)                          % Plot the phase response after
                                        % subtracting.
grid on                                 % Turn on grid for phase plot.
title('Experimental Minus 25/(s^2+2*0.22*5s+5^2)')
                                        % Add title.
xlabel('Frequency (rad/sec)')           % Add x-axis label.
ylabel('Phase deg')                     % Add y-axis label.
'This completes a portion of Example 10.18.'
'The student should continue the program for practice.'
pause
```

Chapter 11: Design Via Frequency Response

ch11p1 (Example 11.1) We can design via gain adjustment on the Bode plot using MATLAB. You will input the desired percent overshoot from the keyboard. MATLAB will calculate the required phase margin and then search the Bode

plot for that phase margin. The magnitude at the phase-margin frequency is the reciprocal of the required gain. MATLAB will then plot a step response for that gain. Let us look at Example 11.1 in the text.

```
'(ch11p1) Example 11.1'              % Display label.
clf                                  % Clear graph on screen.
numg=[100];                          % Define numerator of G(s).
deng=poly ([0 -36 -100]);            % Define denominator of G(s).
G=tf (numg, deng)                    % Create and display G(s).
pos=input ('Type %OS');              % Input desired percent overshoot.
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2));
                                     % Calculate required damping ratio.
Pm=atan(2*z/(sqrt(-2*z^2+sqrt(l+4*z^4))))*(180/pi);
                                     % Calculate required phase margin.
w=0.01:0.01:1000;                    % Set range of frequency from 0.01
                                     % to 1000 in steps of 0.01.
[M,P]=bode(G,w);                     % Get Bode data.
Ph=-180+Pm;                          % Calculate required phase angle.
for k=1: 1: length (P);              % Search Bode data for required
                                     % phase angle.
if P(k)-Ph <=0;                      % If required phase angle is found,
                                     % find the value of
M=M(k);                              % magnitude at the same frequency.
'Required K'                         % Display label.
K=1/M                                % Calculate the required gain.
break                                % Stop the loop.
end                                  % End if.
end                                  % End for.
T=feedback(K*G,1);                   % Find T(s) using the calculated K.
step(T)                              % Generate a step response.
title (['Closed-Loop Step Response for K=',num2str (K)])
                                     % Add title to step response.
pause
```

ch11p2 (Example 11.2) Let us use MATLAB to design a lag compensator. The program solves Example 11.2 in the text and follows the same design technique demonstrated in that example. You will input the value of gain to meet the steady-state error requirement followed by the desired percent overshoot. MATLAB then designs a lag compensator, evaluates K_v, and generates a closed-loop step response.

```
'(ch11p2) Example 11.2'              % Display label.
clf                                  % Clear graph on screen.
K=input('Type value of K to meet steady-state error requirement');
                                     % Input K.
pos=input ('Type %OS');              % Input desired percent overshoot.
numg=[100*K];                        % Define numerator of G(s).
deng=poly ([0-36-100]);              % Define denominator of G(s).
'G(s)'                               % Display label.
G=tf (numg, deng)                    % Create and display G(s).
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2));
                                     % Calculate required damping
                                     % ratio.
Pm=atan(2*z/(sqrt(-2*z^2+sqrt(l+4*z^4))))*(180/pi)+10;
```

```
                                    % Calculate required phase margin.
w=0.01:0.01:100;                    % Set range of frequency from 0.01
                                    % to 1000 in steps of 0.01.
[M,P]=bode(G,w);                    % Get Bode data.
Ph=-180+Pm;                         % Calculate required phase angle.
for k=1: 1: length (P);             % Search Bode data for required
                                    % phase angle.
if P (k) -Ph <=0;                   % If required phase angle is found,
                                    % find the value of
M=M(k);                             % magnitude at the same frequency.
wf=w(k);                            % At this frequency the magnitude
                                    % plot must go through 0 dB.
break                               % Stop the loop.
end                                 % End if.
end                                 % End for.
wh=wf/10;                           % Calculate the high-frequency
                                    % break of the lag compensator.
wl=(wh/M);                          % Calculate the low-frequency
                                    % break of the lag compensator;
                                    % found from lag compensator,
                                    % Gc(s)=Kc(s+wh)/(s+wl), high & low
                                    % frequency gain requirements.
                                    % At low w, gain=1. Thus,
                                    % Kc*wh/wl=1. At high w, gain=1/M.
                                    % Thus Kc=1/M. Hence
                                    % Kc=wl/wh=1/M, or wl=wh/M.
numc=[1 wh];                        % Generate numerator of lag
                                    % compensator, Gc(s).
denc=[1 wl];                        % Generate denominator of lag
                                    % compensator, Gc(s).
Kc=wl/wh;                           % Generate K for Gc(s).
'Lag compensator'                   % Display label.
Kc                                  % Display lag compensator K.
'Gc(s)'                             % Display label.
Gc=tf (Kc*numc, denc)               % Create and display Gc(s).
'Gc(s)G(s)'                         % Display label.
GcG=Gc*G                            % Create and display Gc(s)G(s).
s=tf ([1 0], 1);                    % Create transfer function,'s'.
sGcG=s*GcG;                         % Create sGc(s)G(s).
sGcG=minreal(sGcG);                 % Cancel common terms.
Kv=dcgain(sGcG)                     % Evaluate Kv.
T=feedback(GcG,1);                  % Create T(s).
step(T)                             % Generate a closed-loop, lag-
                                    % compensated step response.
title ('Closed-Loop Step Response for Lag-Compensated System')
                                    % Add title to step response.
pause
```

ch11p3 (Example 11.3) Let us use MATLAB to design a lead compensator. The program solves Example 11.3 in the text and follows the same design technique demonstrated in that example. You will enter desired percent overshoot, peak time, and K_v. MATLAB then designs the lead compensator using Bode plots, calculates K_v, and plots a closed-loop step response.

```
'(ch11p3) Example 11.3'              % Display label.
pos=input ('Type %OS ');             % Input desired percent overshoot.
Tp=input('Type peak time ');         % Input desired peak time.
Kv=input('Type value of Kv ');       % Input Kv.
numg=[100];                          % Define numerator of G(s).
deng=poly ([0 -36 -100]);            % Define denominator of G(s).
G=tf(numg,deng);                     % Create G(s).
s=tf([1 0], 1);                      % Create transfer function,'s'.
sG=s*G;                              % Create sG(s).
sG=minreal(sG);                      % Cancel common factors.
K=dcgain(Kv/sG);                     % Solve for K.
'G(s)'                               % Display label.
G=zpk (K*G)                          % Put K into G (s), convert to
                                     % factored form, and display.
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2));
                                     % Calculate required damping
                                     % ratio.
Pm=atan(2*z/(sqrt(-2*z^2+sqrt(l+4*z^4))))*(180/pi);
                                     % Calculate required phase margin.
wn=pi/ (Tp*sqrt(l-z^2));             % Calculate required natural
                                     % frequency.
wBW=wn*sqrt((1-2*z^2)+sqrt(4*z^4-4*z^2+2));
                                     % Determine required bandwidth.
w=0.01:0.5:1000;                     % Set range of frequency from 0.01
                                     % to 1000 in steps of 0.5
[M,P]=bode(G,w);                     % Get Bode data.
[Gm,Pm,Wcg,Wcp]=margin(G);           % Find current phase margin.
Pmreq=atan(2*z/(sqrt(-2*z^2+sqrt(l+4*z^4))))*(180/pi);
                                     % Calculate required phase margin.
Pmreqc=Pmreq+10;                     % Add a correction factor of 10
                                     % degrees.
Pc=Pmreqc-Pm;                        % Calculate phase contribution
                                     % required from lead compensator.
                                     % Design lead compensator
beta=(1-sin(Pc*pi/180))/(1+sin(Pc*pi/180));
                                     % Find compensator beta.
magpc=1/sqrt(beta);                  % Find compensator peak magnitude.
for k=1: 1: length(M);               % Find frequency at which
                                     % uncompensated system has a
                                     % magnitude of 1/magpc.
                                     % This frequency will be the new
                                     % phase margin frequency.
if M(k)-(1/magpc) <=0;               % Look for peak magnitude.
wmax=w(k);                           % This is the frequency at the
                                     % peak magnitude.
break                                % Stop the loop,
end                                  % End if.
end                                  % End for.
% Calculate lead compensator zero, pole, and gain.
zc=wmax*sqrt(beta);                  % Calculate the lead compensator's
                                     % low break frequency.
pc=zc/beta;                          % Calculate the lead compensator's
                                     % high break frequency.
Kc=1/beta;                           % Calculate the lead compensator's
```

```
                                        % gain.
'Gc(s)'                                 % Display label.
Gc=tf (Kc*[1 zc],[1 pc]);               % Create Gc(s).
Gc=zpk(Gc)                              % Convert Gc (s) to factored form
                                        % and display.
'Ge(s)=G(s)Gc(s)'                       % Display label.
Ge=G*Gc                                 % Form Ge(s)=Gc(s)G(s).
sGe=s*Ge;                               % Create sGe(s).
sGe=minreal (sGe);                      % Cancel common factors.
Kv=dcgain(sGe)                          % Calculate Kv.
T=feedback(Ge,1);                       % Find T(s).
step(T)                                 % Generate closed-loop, lead-
                                        % compensated step response.
title('Lead-Compensated Step Response')
                                        % Add title to lead-compensated
                                        % step response.
pause
```

ch11p4 (Example 11.4) Let us use MATLAB to design a lag-lead compensator. The program solves Example 11.4 in the text and follows the same design technique demonstrated in that example. You will enter desired percent overshoot, peak time, and K_v. MATLAB then designs the lag-lead compensator using Bode plots, calculates K_v and plots a closed-loop step response.

```
'(ch11p4) Example 11.4'                 % Display label.
pos=input('Type %OS ');                 % Input desired percent overshoot.
Tp=input('Type peak time ');            % Input desired peak time.
Kv=input('Type value of Kv ');          % Input desired Kv.
numg=[1];                               % Define numerator of G(s).
deng=poly([0 -1 -4]);                   % Define denominator of G(s).
G=tf(numg,deng);                        % Create G(s) without K.
s=tf([1 0],1);                          % Create transfer function,'s'.
sG=s*G;                                 % Create sG(s).
sG=minreal(sG);                         % Cancel common factors.
K=dcgain(Kv/sG);                        % Solve for K.
'G(s)'                                  % Display label.
G=tf(K*numg,deng);                      % Put K into G(s).
G=zpk(G)                                % Convert G (s) to factored form and
                                        % display.
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2));
                                        % Calculate required damping ratio.
Pmreq=atan(2*z/(sqrt(-2*z^2+sqrt(1+4*z^4))))*(180/pi);
                                        % Calculate required phase margin.
wn=pi/(Tp*sqrt(1-z^2));                 % Calculate required natural
                                        % frequency.)
wBW=wn*sqrt((1-2*z^2)+sqrt(4*z^4-4*z^2+2));
                                        % Determine required bandwidth.
wpm=0.8*wBW;                            % Choose new phase-margin
                                        % frequency.
[M,P]=bode(G,wpm);                      % Get Bode data.
Pmreqc=Pmreq-(180+P)+5;                 % Find phase contribution required
                                        % from lead compensator
                                        % with additional 5 degrees.
beta=(1-sin(Pmreqc*pi/180))/(1+sin(Pmreqc*pi/180));
```

```
                                % Find beta.
                                % Design lag compensator zero, pole,
                                % and gain.
zclag=wpm/10;                   % Calculate zero of lag compensator.
pclag=zclag*beta;               % Calculate pole of lag compensator.
Kclag=beta;                     % Calculate gain of lag compensator.
'Lag compensator, Glag(s)'      % Display label.
Glag=tf(Kclag*[1zclag],[1pclag]);  % Create lag compensator.
Glag=zpk(Glag)                  % Convert Glag(s) to factored form
                                % and display.
                                % Design lead compensator zero,
                                % pole, and gain.
zclead=wpm*sqrt(beta);          % Calculate zero of lead
                                % compensator.
pclead=zclead/beta;             % Calculate pole of lead
                                % compensator.
Kclead=1/beta;                  % Calculate gain of lead
                                % compensator.
'Lead compensator'              % Display label.
Glead=tf(Kclead*[1 zclead],[1 pclead]);
                                % Create lead compensator.
Glead=zpk(Glead)                % Convert Glead(s) to factored form
                                % and display.
'Lag-Lead Compensated Ge(s)'    % Display label.
Ge=G*Glag*Glead                 % Create compensated system,
                                % Ge(s)=G(s) Glag(s) Glead(s).
sGe=s*Ge;                       % Create sGe(s).
sGe=minreal (sGe);              % Cancel common factors.
Kv=dcgain(sGe)                  % Calculate Kv
T=feedback(Ge,1);               % Find T(s).
step(T)                         % Generate closed-loop, lag-lead-
                                % compensated step response.
title('Lag-Lead-Compensated Step Response')
                                % Add title to lag-lead-
                                % compensated
                                % step response.
pause
```

Chapter 12: Design Via State Space

ch12p1 (Example 12.1) We can use MATLAB to design controller gains using pole placement. You will enter the desired percent overshoot and settling time. We introduce the following commands: `[num, den]=ord2 (wn, z)`, which produces a second-order system, given the natural frequency `(wn)` and the damping ratio `(z)`. Then we use the denominator `(den)` to specify the dominant poles; and `K=acker (A, B, -poles)`, which calculates controller gains from the system matrix `(A)`, the input matrix `(B)`, the desired poles `(poles)`. Let us look at Example 12.1 in the text.

```
'(ch12p1) Example 12.1'         % Display label.
clf                             % Clear graph on screen.
numg=2 0*[1 5];                 % Define numerator of G(s).
deng=poly ([0 -1 -4]);          % Define denominator of G(s).
'Uncompensated G(s)'            % Display label.
G=tf (numg, deng)               % Create and display G(s).
```

```
pos=input('Type desired %OS ');      % Input desired percent overshoot.
Ts=input('Type desired settling time');
                                     % Input desired settling time.
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2));
                                     % Calculate required damping ratio.
wn=4/ (z*Ts);                        % Calculate required natural
                                     % frequency.
[num,den]=ord2 (wn, z);              % Produce a second-order system
                                     % that meets the transient response
                                     % requirements.
r=roots(den);                        % Use denominator to specify
                                     % dominant poles.
poles=[r(1)r(2)-5.1];                % Specify pole placement for all
                                     % poles.
characteristiceqdesired=poly(poles)
                                     % Form desired characteristic
                                     % polynomial for display.
[Ac Bc Cc Dc]=tf 2ss (numg,deng);    % Find controller canonical form
                                     % of state-space representation
                                     % of G(s).
P=[0 0 1; 0 1 0; 1 0 0];             % Transformation matrix for
                                     % controller canonical to phase-
                                     % variable form.
Ap=inv(P)*Ac*P;                      % Transform Ac to phase-variable
                                     % form.
Bp=inv(P)*Bc;                        % Transform Bc to phase-variable
                                     % form.
Cp=Cc*P;                             % Transform Cc to phase-variable
                                     % form.
Dp=Dc;                               % Transform Dc to phase-variable
                                     % form.
Kp=acker(Ap,Bp,poles)                % Calculate controller gains in
                                     % phase-variable form.
Apnew=Ap-Bp*Kp;                      % Form compensated A matrix.
Bpnew=Bp;                            % Form compensated B matrix.
Cpnew=Cp;                            % Form compensated C matrix.
Dpnew=Dp;                            % Form compensated D matrix.
[numt,dent]=ss2tf(Apnew,Bpnew,Cpnew,Dpnew);
                                     % Form T(s) numerator and
                                     % denominator.
'T(s)'                               % Display label.
T=tf (numt, dent)                    % Create and display T(s).
poles=roots (dent)                   % Display poles of T(s).
Tss=ss (Apnew, Bpnew,Cpnew,Dpnew)    % Create and display Tss, an LTI
                                     % state-space object.
step (Tss)                           % Produce compensated step
                                     % response.
title('Compensated Step Response')
                                     % Add title to compensated step
                                     % response.
pause
```

ch12p2 (Example 12.2) We can test controllability by using the MATLAB command `Cm=ctrb(A,B)` to find the controllability matrix given the system matrix

(A) and the input matrix (B). This command is followed by rank (Cm) to test the rank of the controllability matrix (Cm). Let us apply the commands to Example 12.2.

```
'(ch12p2) Example 12.2'              % Display label.
A=[-1 1 0;0 -1 0;0 0 -2]              % Define compensated A matrix.
B=[0;1;1]                            % Define compensated B matrix.
Cm=ctrb(A,B)                         % Calculate controllability
                                     % matrix.
Rank=rank(Cm)                        % Find rank of controllability
                                     % matrix.
pause
```

ch12p3 (Example 12.4) If we design controller gains using MATLAB, we do not have to convert to phase-variable form. MATLAB will give us the controller gains for any state-space representation we input. Let us look at Example 12.4 in the text.

```
'(ch12p3) Example 12.4'              % Display label.
clf                                  % Clear graph on screen.
A=[-5 1 0;0 -2 1;0 0 -1];            % Define system matrix A.
B=[0; 0; 1];                         % Define input matrix B.
C=[-110];                            % Define output matrix C.
D=0;                                 % Define matrix D.
pos=input('Type desired %OS ');      % Input desired percent overshoot.
Ts=input('Type desired settling time')
                                     % Input desired settling time.
z=(-log(pos/100))/(sqrt(pi^2+log(pos/100)^2));
                                     % Calculate required damping ratio.
wn=4/(z*Ts);                         % Calculate required natural
                                     % frequency.
[num,den]=ord2(wn, z);               % Produce a second-order system
                                     % that meets the transient
                                     % requirements.
r=roots(den);                        % Use denominator to specify
                                     % dominant poles.
poles=[r(1)r(2)-4];                  % Specify pole placement for all
                                     % poles.
K=acker(A,B,poles)                   % Calculate controller gains.
Anew=A-B*K;                          % Form compensated A matrix.
Bnew=B;                              % Form compensated B matrix.
Cnew=C;                              % Form compensated C matrix.
Dnew=D;                              % Form compensated D matrix.
Tss=ss(Anew,Bnew,Cnew,Dnew);         % Form LTI state-space object.
'T(s)'                               % Display label.
T=tf(Tss);                           % Create T(s).
T=minreal(T)                         % Cancel common terms and display
                                     % T(s).
poles=pole(T)                        % Display poles of T(s).
step(Tss)                            % Produce compensated step
                                     % response.
title('Compensated Step Response')
                                     % Add title to compensated step
                                     % response.
pause
```

ch12p4 (Example 12.5) We can design observer gains by using the command `l=acker(A',C',poles)'`. Notice we use the transpose of the system matrix `(A)` and output matrix `(C)` along with the desired poles `(poles)`. Let us look at Example 12.5 in the text.

```
'(ch12p4) Example 12.5'                 % Display label.
numg=[14];                              % Define numerator of G(s).
deng=poly([-1 -2 -5]);                  % Define denominator of G(s).
'G(s)'                                  % Display label.
G=tf(numg,deng)                         % Create and display G(s).
[Ac,Bc,Cc,Dc]=tf2ss(numg,deng);         % Transform G(s) to controller
                                        % canonical form in state space.
Ao=Ac';                                 % Transform Ac to observer
                                        % canonical form.
Bo=Cc';                                 % Transform Bc to observer
                                        % canonical form.
Co=Bc';                                 % Transform Cc to observer
                                        % canonical form.
Do=Dc;                                  % Transform Dc to observer
                                        % canonical form.
r=roots([1 2 5])                        % Find the controller-compensated
                                        % system poles.
poles=10*[r' 10*real(r(1))]             % Make observer poles 10x bigger.
lp=acker(Ao',Co',poles)'                % Find the observer gains in
                                        % observer canonical form.
pause
```

ch12p5 (Example 12.6) We can test observability using the MATLAB command `Om=obsv(A, C)` to find the observability matrix given the system matrix `(A)` and the output matrix `(C)`. This command is followed by `rank(Om)` to test the rank of the observability `matrix (Om)`. Let us apply the commands to Example 12.6.

```
'(ch12p5) Example 12.6'                 % Display label.
A=[0 1 0;0 0 1;-4 -3 -2]                % Define compensated A matrix.
C=[0 5 1]                               % Define compensated C matrix.
Om=obsv(A,C)                            % Form observability matrix.
Rank=rank(Om)                           % Find rank of observability
                                        % matrix.
pause
```

ch12p6 (Example 12.8) We can design observer gains using the command `l=acker (A', C', poles)'` without transforming to observer canonical form. Let us look at Example 12.8 in the text.

```
'(ch12p6) Example 12.8'                 % Display label.
A=[-5 1 0;0 -2 1;0 0 -1];               % Define system matrix A.
B=[0;0;1];                              % Define input matrix B.
C=[1 0 0];                              % Define output matrix C.
D=0;                                    % Define matrix D.
poles=roots([1 120 2500 50000])         % Specify pole placement for all
                                        % poles.
l=acker(A', C',poles)'                  % Calculate observer gains.
pause
```

Chapter 13: Digital Control Systems

ch13p1 (Example 13.4) We can convert $G_1(s)$ in cascade with a zero-order `hold` (z.o.h.) to $G(z)$ using MATLAB's `G=c2d (G1, T,'zoh')` command, where `G1` is an LTI continuous-system object and `G` is an LTI sampled-system object. `T` is the sampling interval and `'zoh'` is a method of transformation that assumes $G_1(s)$ in cascade with a z.o.h. We simply put $G_1(s)$ into the command (the z.o.h. is automatically taken care of) and the command returns $G(z)$. Let us apply the concept to Example 13.4. You will enter `T` through the keyboard.

```
'(ch13p1) Example 13.4'              % Display label.
T=input('Type T ');                  % Input sampling interval.
numg1s=[1 2];                        % Define numerator of G1(s).
deng1s=[1 1];                        % Define denominator of G1(s).
'G1(s)'                              % Display label.
G1=tf(numg1s,deng1s)                 % Create G1(s) and display.
'G(z)'                               % Display label.
G=c2d(G1,T,'zoh')                    % Convert G1 (s) in cascade with
                                     % z.o.h. to G(z) and display.
pause
```

ch13p2 We also can use MATLAB to convert $G(s)$ to $G(z)$ when $G(s)$ is not in cascade with a z.o.h. The command `H=c2d(F, T,' zoh')` transforms $F(s)$ in cascade with a z.o.h. to $H(z)$, where $H(z) = ((z-1)/z)^* z\{F(s)/s\}$. If we let $F(s) = sG(s)$, the command solves for $H(z)$, where $H(z) = ((z-1)/z)^* z\{G(s)\}$. Hence, $z\{G(s)\} = (z/[z-1])^* H(z)$. In summary, input $F(s) = sG(s)$, and multiply the result of `H=c2d (F, T,'zoh')` by $(z/[z-1])$. This process is equivalent to finding the z-transform. We convert $G(s) = (s+3)/(s^2+6s+13)$ into $G(z)$. You will enter `T`, the sampling interval, through the keyboard. `T` is used to form $H(z)$. We use an unspecified sampling interval, `T=[]`, to form $z/(z-1)$.

```
'(ch13p2)'                           % Display label.
T=input('Type T ');                  % Input sampling interval.
numgs=[1 3];                         % Define numerator of G(s).
dengs=[1 6 13];                      % Define denominator of G(s).
'G(s)'                               % Display label.
Gs=tf(numgs,dengs)                   % Create and display G(s).
Fs=Gs*tf([1 0],1)                    % Create F(s)=sG(s).
Fs=minreal(Fs);                      % Cancel common poles and zeros.
Hz=c2d(Fs,T,'zoh');                  % Convert F(s) to H(z) assuming
                                     % z.o.h.
Gz=Hz*tf([1 0],[1-1],[]);            % Form G(z)=H(z)*z/(z-1).
'G(z)'                               % Display label.
Gz=minreal(Gz)                       % Cancel common poles and zeros.
Pause
```

ch13p3 Creating Digital Transfer Functions Directly Vector Method, Polynomial Form

A digital transfer function can be expressed as a numerator polynomial divided by a denominator polynomial, that is, $F(z) = N(z)/D(z)$. The numerator, $N(z)$, is represented by a vector, `numf`, that contains the coefficients of $N(z)$. Similarly, the denominator, $D(z)$, is represented by a vector, `denf`, that contains the coefficients of $D(z)$. We form $F(z)$ with the command, `F=tf(numf, denf, T)`, where `T` is the sampling interval. `F` is called a linear time-invariant (LTI) object. This object, or

transfer function, can be used as an entity in other operations, such as addition or multiplication. We demonstrate with $F(z) = 150(z^2 + 2z + 7)/(z^2 - 0.3z + 0.02)$. We use an unspecified sampling interval, T = [ ]. Notice after executing the `tf` command, MATLAB prints the transfer function.

Vector Method, Factored Form

We also can create digital LTI transfer functions if the numerator and denominator are expressed in factored form. We do this by using vectors containing the roots of the numerator and denominator. Thus, $G(s) = K^* N(z)/D(z)$ can be expressed as an LTI object using the command, `G=zpk (numg, deng, K, T)`, where `numg` is a vector containing the roots of $N(z)$, `deng` is a vector containing the roots of $D(z)$, `K` is the gain, and `T` is the sampling interval. The expression `zpk` stands for zeros (roots of the numerator), poles (roots of the denominator), and gain, `K`. We demonstrate with $G(z) = 20(z + 2)(z + 4)/[(z - 0.5)(z - 0.7)(z - 0.8)]$ and an unspecified sampling interval. Notice after executing the `zpk` command, MATLAB prints the transfer function.

Rational Expression in *z* Method, Polynomial Form (Requires Control System Toolbox 8)

This method allows you to type the transfer function as you normally would write it. The statement `z=tf('z')` must precede the transfer function if you wish to create a digital LTI transfer function in polynomial form equivalent to using `G=tf(numg, deng,T)`.

Rational Expression in *z* Method, Factored Form (Requires Control System Toolbox 8)

This method allows you to type the transfer function as you normally would write it. The statement `z=zpk('z')` must precede the transfer function if you wish to create a digital LTI transfer function in factored form equivalent to using `G=zpk (numg, −deng,K,T)`.

For both rational expression methods the transfer function can be typed in any form regardless of whether `z=tf('z') or z=zpk('z')` is used. The difference is in the created digital LTI transfer function. We use the same examples above to demonstrate the rational expression in z methods.

```
'(ch13p3)'                          % Display label.
'Vector Method, Polynomial Form'    % Display label.
numf=150*[1 2 7]                    % Store 150(z^2+2z+7) in numf and
                                    % display.
denf=[1 -0.3 0.02]                  % Store(z^2-0.3z+0.02) in denf and
                                    % display.
'F(z)'                              % Display label.
F=tf(numf ,denf,[])                 % Form F(z) and display.
clear                               % Clear previous variables from
                                    % workspace.
'Vector Method, Factored Form'      % Display label.
numg=[-2 -4]                        % Store (s+2) (s+4) in numg and
                                    % display.
deng=[0.5 0.70.8]                   % Store (s-0.5)(s-0.7)(s-0.8) in
                                    % deng and display.
K=20                                % Define K.
```

```
'G(z)'                                  % Display label.
G=zpk(numg,deng,K,[])                   % Form G(z) and display,
clear                                   % Clear previous variables from
                                        % workspace.
'Rational Expression Method, Polynomial Form'
                                        % Display label.
z=tf('z')                               % Define z as an LTI object in
                                        % polynomial form.
F=150*(z^2+2*z+7)/(z^2-0.3*z+0.02)
                                        % Form F(z) as an LTI transfer
                                        % function in polynomial form.
G=20*(z+2)*(z+4)/[(z-0.5)*(z-0.7)*(z-0.8)]
                                        % Form G(z) as an LTI transfer
                                        % function in polynomial form.
clear                                   % Clear previous variables from
                                        % workspace.
'Rational Expression Method, Factored Form'
                                        % Display label.
z=zpk('z')                              % Define z as an LTI object in
                                        % factored form.
F=150*(z^2+2*z+7)/(z^2-0.3*z+0.02)
                                        % Form F(z) as an LTI transfer
                                        % function in factored form.
G=20*(z+2)*(z+4)/[(z-0.5)*(z-0.7)*(z-0.8)]
                                        % Form G(z) as an LTI transfer
                                        % function in factored form,
pause
```

ch13p4 We also can use MATLAB to convert $G(z)$ to $G(s)$ when $G(s)$ is not in cascade with a z.o.h. First, we create a sampled LTI transfer function, as discussed in ch13p3. The command `F=d2c(H,'zoh')` transforms $H(z)$ to $F(s)$ in cascade with a z.o.h., where $H(z) = ((z-1)/z)^* z\{F(s)/s\}$. If we consider $F(s) = sG(s)$, the command solves for $sG(s)$ given $H(z)$. Finally, $sG(s)/s = G(s)$ yields the final result. In summary, form $H(z)$, where $H(z) = ((z-1)/z)^* G(z)$. Use `F=d2c(H,'zoh')` to find $F(s) = sG(s)$. Divide the result by s and obtain $G(s)$. We convert $G(z) = z/(z-0.3)$ into $G(s)$. You will enter `T`, the sampling interval, through the keyboard.

```
'(ch13p4)'                              % Display label.
T=input('Type T');                      % Input sampling interval.
numgz=[10];                             % Define numerator of G(z).
dengz=[1 -.3];                          % Define denominator of G(z).
'G(z)'                                  % Display label.
Gz=tf(numgz,dengz,T).                   % Create and display G(z).
Hz=Gz*tf([1 -1],[1 0],T);               % Create H(z)=((z-1)/z)*G(z).
Hz=minreal(Hz);                         % Cancel common poles and zeros.
Fs=d2c(Hz,'zoh');                       % Convert from H(z) to F(s)=sG(s).
Gs=Fs*tf(1,[10]);                       % Create G(s)=F(s)(1/s).
'G(s)'                                  % Display label.
Gs=minreal(Gs)                          % Cancel common poles and zeros.
pause
```

ch13p5 (Example 13.6) We can use MATLAB to find the gain for stability. Let us look at Example 13.6 in the text.

```
'(ch13p5) Example 13.6'          % Display label.
numgas=27;                       % Define numerator of Ga(s).
dengas=[1 27 0];                 % Define denominator of Ga(s).
'Ga(s)'                          % Display label.
Ga=tf(numgas,dengas).            % Create and display Ga(s).
'G(z)'                           % Display label.
Gz=c2d(Ga,0.1,'zoh')             % Find G(z) assuming Ga(s) in
                                 % cascade with z.o.h. and display.
for K=1:0.1:50;                  % Set range of K to look for
                                 % stability.
Tz=feedback(K*Gz,1);             % Find T(z).
r=pole(Tz);                      % Get poles for this value of K.
rm=max(abs(r));                  % Find pole with maximum absolute
                                 % value for this value of K.
if rm>=1,                        % See if pole is outside unit
                                 % circle.
break;                           % Stop if pole is found outside
                                 % unit circle.
end;                             % End if.
end;                             % End for.
K                                % Display K value.
r                                % Display closed-loop poles for
                                 % this value of K.
rm                               % Display absolute value of pole.
pause
```

ch13p6 (Example 13.9) We can use MATLAB's command `dcgain(Gz)` to find steady-state errors. The command evaluates the dc gain of `Gz`, a digital LTI transfer function object, by evaluating `Gz` at $z = 1$. We use the dc gain to evaluate, K_p, K_v, and K_a. Let us look at Example 13.9 in the text. You will input `T`, the sampling interval, through the keyboard to test stability.

```
'(ch13p6) Example 13.9'          % Display label.
T=input('Type T');               % Input sampling interval.
numg1s=[10];                     % Define numerator of G1(s).
deng1s=poly([0 -1]);             % Define denominator of G1(s).
'G1(s)'                          % Display label.
G1s=tf(numg1s,deng1s)            % Create and display G1(s).
'G(z)'                           % Display label.
Gz=c2d(G1s,T,'zoh')              % Convert G1(s) and z . o. h. to G(z)
                                 % and display.
'T(z)'                           % Display label.
Tz=feedback(Gz, 1)               % Create and display T(z).
'Closed-Loop z-Plane Poles'      % Display label.
r=pole(Tz)                       % Check stability.
M=abs(r)                         % Display magnitude of roots.
pause
Kp=dcgain(Gz)                    % Calculate Kp.
GzKv=Gz*(1/T)*tf([1-1],[10],T);
                                 % Multiply G(z) by (1/T)*(z-1).
                                 % Also, divide G(z) by z , which
                                 % makes transfer function proper
                                 % and yields same Kv.
GzKv=mineral (GzKv, 0.00001);    % Cancel common poles and zeros.
Kv=dcgain(GzKv)                  % Calculate Kv.
```

```
GzKa=Gz*(1/T^2)*tf([1-2 1],[100],T);
                                    % Multiply G(z) by (1/T"2)(z-1)^2.
                                    % Also, divide G(z)by z^2 , which
                                    % makes the transfer function
                                    % proper and yields the same Ka.
GzKa=minreal (GzKa, 00.0001)        % Cancel common poles and zeros.
Ka=dcgain(GzKa)                     % Calculate Ka.
pause
```

ch13p7 (Example 13.10) We now use the root locus to find the gain for stability. First, we create a digital LTI transfer-function object for $G(z) = N(z)/D(z)$, with an unspecified sampling interval. The LTI object is created using `tf (numgz, dengz,[])`, where `numgz` represents $N(z)$,`dengz` represents $D(z)$, and `[]` indicates an unspecified sampling interval. MATLAB produces a z-plane root locus along with the unit circle superimposed using the command, `zgrid ([],[])`. We then interactively select the intersection of the root locus and the unit circle. MATLAB responds with the value of gain and the closed-loop poles. Let us look at Example 13.10.

```
'(ch13p7) Example 13.10'            % Display label.
clf                                 % Clear graph.
numgz=[11];                         % Define numerator of G(z).
dengz=poly([1 0.5]);                % Define denominator of G(z).
'G(z)'                              % Display label.
Gz=tf(numgz,dengz,[]).              % Create and display G(z).
rlocus(Gz)                          % Plot root locus.
zgrid([],[])                        % Add unit circle to root locus.
title (['z-Plane Root Locus'])      % Add title to root locus.
[K,p]=rlocfind(Gz)                  % Allows input of K by selecting
                                    % point on graphic.
pause
```

ch13p8 (Example 13.11) We now use the root locus to find the gain to meet a transient response requirement. After MATLAB produces a z-plane root locus, along with damping ratio curves superimposed using the command `zgrid`, we interactively select the desired operating point at a damping ratio of 0.7, thus determining the gain. MATLAB responds with a gain value as well as the step response of the closed-loop sampled system using `step (Tz)`, where `Tz` is a digital LTI transfer-function object. Let us look at Example 13.11.

```
'(ch13p8) Example 13.11'            % Display label.
clf                                 % Clear graph.
numgz=[11];                         % Define numerator of G(z).
dengz=poly([1 0.5]);                % Define denominator of G(z).
'G(z)'                              % Display label.
Gz=tf(numgz,dengz,[]).              % Create and display G(z).
rlocus(Gz)                          % Plot root locus.
axis([0,1,-1,1])                    % Create close-up view.
zgrid                               % Add damping ratio curves to root
                                    % locus.
title(['z-Plane Root Locus'])       % Add title to root locus.
[K,p]=rlocfind(Gz)                  % Allows input of K by selecting
                                    % point on graphic.
'T(z)'                              % Display label.
Tz=feedback(K*Gz,1)                 % Find T(z).
```

```
step(Tz)                                % Find step response of gain-
                                        % compensated system.
title (['Gain Compensated Step Response'])
                                        % Add title to step response of
                                        % gain-compensated system.
pause
```

ch13p9 (Example 13.12) Let us now use MATLAB to design a digital lead compensator. The *s*-plane design was performed in Example 9.6. Here we convert the design to the *z*-plane and run a digital simulation of the step response. Conversion of the *s*-plane lead compensator, $Gc(s)$=`numgcs/dengcs`, to the *z*-plane compensator, $Gc(z)$=`numgcz/dengcz`, is accomplished using the `Gcz=c2d (numgcs, dengcs, T, 'tustin')` command to perform a Tustin transformation, where `T`=sampling interval, which for this example is 1/300. This exercise solves Example 13.12 using MATLAB.

```
'(ch13p9) Example 13.12'                % Display label.
clf                                     % Clear graph.
T=0.01                                  % Define sampling interval.
numgcs=1977*[16];                       % Define numerator of Gc(s).
dengcs=[1 29.1];                        % Define denominator of Gc(s).
'Gc(s) in polynomial form'              % Print label.
Gcs=tf(numgcs,dengcs)                   % Create Gc(s) in polynomial form
                                        % and display.
'Gc(s) in polynomial form'              % Display label.
Gcszpk=zpk(Gcs)                         % Create Gc(s) in factored form
                                        % and display.
'Gc(z) in polynomial form via Tustin Transformation'
                                        % Display label.
Gcz=c2d(Gcs,T,'tustin')                 % FormGc(z) via Tustin
                                        % transformation.
'Gc(z) in factored form via Tustin Transformation'
                                        % Display label.
Gczzpk=zpk(Gcz)                         % Show Gc(z) in factored form.
numgps=1                                % Define numerator of Gp(s).
dengps=poly([0 -6 -10]);                % Define denominator of Gp(s).
'Gp(s) in polynomial form               % Display label.
Gps=tf(numgps,dengps)                   % Create Gp(s) in polynomial form
                                        % and display.
'Gp(s) in factored form'                % Display label.
Gpszpk=zpk(Gps)                         % Create Gp(s) in factored form
                                        % and display.
'Gp(z) in polynomial form'              % Display label.
Gpz=c2d(Gps,T,'zoh')                    % Form Gp(z) via zoh trans formation.
'Gp(z) in factored form'                % Display label.
Gpzzpk=zpk(Gpz)                         % Form Gp(z) in factored form.
Gez=Gcz*Gpz                             % Form Ge(z)=Gc(z) Gp(z).
'Ge(z)=Gc(z)Gp(z) in factored form'
                                        % Display label.
Gezzpk=zpk(Gez)                         % Form Ge(z) in factored form
                                        % and display.
'z-1'                                   % Display label.
zm1=tf([1 -1],1,T)                      % Form z-1.
zm1Gez=mineral (zm1*Gez , 00.0001);     % Cancel common factors.
```

```
'(z-1)Ge(z) for finding steady-state error'
                                    % Display label.
zm1Gezzpk=zpk(zm1Gez)               %Form & display(z-1)Ge(z) in
                                    % factored form.
Kv=(1/T)*dcgain(zm1Gez)             % Find Kv.
'T(z)=Ge(z)/(1+Ge(z))'              % Display label.
Tz=feedback(Gez , 1)                % Find closed-loop
                                    % transfer function, T(z)
step(Tz, 0:T:2)                     % Find step reponse.
title('Closed-Loop Digital Lead Compensated Step Response')
                                    % Add title to step response.
```

B.3 Command Summary

`abs(x)`	Obtain absolute value of x.
`acker(A,B,poles)`	Find gains for pole placement.
`angle(x)`	Compute the angle of x in radians.
`atan(x)`	Compute arctan(x).
`axis([xmin,xmax,ymin,ymax])`	Define range on axes of a plot.
`bode(G,w)`	Make a Bode plot of transfer function $G(s)$ over a range of frequencies, ω.
	Field ω is optional.
`break`	Exit loop.
`c2d(G,T,'tustin')`	Convert $G(s)$ to $G(z)$ using the Tustin transformation. T is the sampling interval.
`c2d(G,T,'zoh')`	Convert $G(s)$ in cascade with a zero-order hold to $G(z)$. T is the sampling interval.
`canon(S,'modal')`	Convert an LTI state-space object, S, to parallel form.
`clear`	Clear variables from workspace.
`clf`	Clear current figure.
`conv([a b c d],[e f g h])`	Multiply $(as^3 + bs^2 + cs + d)$ by $(es^3 + fs^2 + gs + h)$.
`ctrb(A,B)`	Find controllability matrix.
`d2c(G,'zoh')`	Convert $G(z)$ to $G(s)$ in cascade with a zero-order hold.
`dcgain(G)`	Find dc gain for $G(s)$ (that is, $s = 0$), or $G(z)$ (that is, $z = 1$).
`eig(A)`	Find eigenvalues of matrix **A**.
`end`	End the loop.
`exp(a)`	Obtain e^a.
`feedback(G, H, sign)`	Find $T(s) = G(s)/[1 \pm G(s)H(s)]$. Sign $= -1$ or is optional for negative feedback systems.
	Sign $= +1$ for positive feedback systems.
`grid on`	Put grid lines on a graph.
`hold off`	Turn off graph hold; start new graph.
`imag(P)`	Form a matrix of the imaginary parts of the components of matrix **P**.

`input('str')`	Permit variable values to be entered from the keyboard with prompt `str`.
`interp1 (x,y,x1)`	Perform table lookup by finding the value of y at the value of $x = x_1$.
`inv(P)`	Find the inverse of matrix **P**.
`length(P)`	Obtain dimension of vector **P**.
`log(x)`	Compute natural log of x.
`log10(x)`	Compute log to the base 10 of x.
`margin(G)`	Find gain and phase margins, and gain and phase margin frequencies of transfer function, $G(s)$.
	Return [Gain margin, Phase margin, 180° frequency, 0 dB frequency].
`max(P)`	Find the maximum component of **P**.
`minreal(G, tol)`	Cancel common factors from transfer function $G(s)$ within tolerance, tol.
	If 'tol' field is blank, a default value is used.
`ngrid`	Superimpose grid over a Nichols plot.
`nichols(G,w)`	Make a Nichols plot of transfer function $G(s)$ over a range of frequencies, ω.
	Field ω is optional.
`nyquist(G,w)`	Make a Nyquist diagram of transfer function $G(s)$ over a range of frequencies, ω
	Field ω is optional.
`obsv(A.C)`	Find observability matrix.
`ord2(wn,z)`	Create a second-order system, $G(s) = 1/[s^2 + 2\zeta\omega_n s + \omega_n^2]$.
`pade(T,n)`	Obtain nth order Padé approximation for delay, T.
`pause`	Pause program until any key is pressed.
`plot(t1,y1,t2,y2,t3,y3)`	Plot $y1$ versus $t1$, $y2$ versus $t2$, and $y3$ versus $t3$ on the same graph.
`pole(G)`	Find poles of LTI transfer function object, $G(s)$.
`poly([-a -b -c])`	Form polynomial $(s + a)(s + b)(s + c)$.
`polyval(P,a)`	Find polynomial $P(s)$ evaluated at a, that is, $P(a)$.
`rank(A)`	Find rank of matrix **A**.
`real(P)`	Form a matrix of the real parts of the components of matrix **P**.
`residue(numf, denf)`	Find residues of $F(s)$=numf/denf.
`rlocfind(GH)`	Allow interactive selection of points on a root locus plot for loop gain, $G(s)H(s)$.
	Return value for K and all closed-loop poles at that K.
`rlocus(GH,K)`	Plot root locus for loop gain, $G(s)H(s)$, over a range of gain, K. The K field is optional.
`roots(P)`	Find roots of polynomial, P.
`semilogx(w,P1)`	Make a semilog plot of P_1 versus $\log_{10}(\omega)$.
`series(G1,G2)`	Find $G_1(s)G_2(s)$.

`sgrid(z,wn)`	Overlay $z(\zeta)$ and `wn`(ω_n) grid lines on a root locus.
`sin(x)`	Find $\sin(x)$.
`sqrt(a)`	Compute $\sqrt{a}$.
`ss2tf(A,B,C,D,1)`	Convert a state-space representation to a transfer function. Return [num, den].
`ss(A,B,C,D)`	Create an LTI state-space object, *S*.
`ss(G)`	Convert an LTI transfer function object, $G(s)$, to an LTI state-space object.
`ssdata(S)`	Extract **A**, **B**, **C**, and **D** matrices from LTI state-space object, *S*.
`step(G1,G2,..Gn,t)`	Plot step responses of $G_1(s)$ through $G_n(s)$ on one graph over a range of time, *t*. Field *t* is optional as are fields G_2 through G_n.
`subplot(xyz)`	Divide plotting area into an *x* by *y* grid with *z* as the window number for the current plot.
`tan(x)`	Find tangent of *x* radians.
`text(a,b,'str')`	Put `str` on graph at graph coordinates, $x = a$, $y = b$.
`tf2ss(numg,deng)`	Convert $G(s) =$ `numg/deng` to state space in controller canonical form. Return `[A, B, C, D]`.
`tf2zp(numg,deng)`	Convert $G(s) =$ `numg/deng` in polynomial form to factored form. Return `[zeros, poles, gains]`.
`tf(numg,deng,T)`	Create an LTI transfer function, $G(s) =$ `numg/deng`, in polynomial form. *T* is the sampling interval and should be used only if *G* is a sampled transfer function.
`tf(G)`	Convert an LTI transfer function, $G(s)$, to polynomial form.
`tfdata(G,'v')`	Extract numerator and denominator of an LTI transfer function, $G(s)$, and convert values to a vector. Return `[num, den]`.
`title('str')`	Put title `str` on graph.
`xlabel('str')`	Put label `str` on *x* axis of graph.
`ylabel('str')`	Put label `str` on *y* axis of graph.
`zgrid`	Superimpose $z(\zeta)$ and `wn` (ω_n) grid curves on a *z*-plane root locus.
`zgrid([],[])`	Superimpose the unit circle on a z-plane root locus.
`zp2tf ([-a -b]',[-c -d]', K)`	Convert $F(s) = K(s+a)(s+b)/(s+c)(s+d)$ to polynomial form. Return `[num, den]`.
`zpk(numg,deng,K,T)`	Create an LTI transfer function, $G(s) =$ `numg/deng`, in factored form. *T* is the sampling interval and should be used only if *G* is a sampled transfer function.
`zpk(G)`	Convert an LTI transfer function, $G(s)$, to factored form.

Bibliography

Johnson, H. et al. *Unmanned Free-Swimming Submersible (UFFS) System Description*. NRL Memorandum Report 4393. Naval Research Laboratory, Washington, DC, 1980.

The MathWorks. *Control System Toolbox™ 8 Getting Started Guide*. The MathWorks, Natick, MA. 2000–2009.

The MathWorks. *Control System Toolbox™ 8 User's Guide*. The MathWorks, Natick, MA. 2001–2009.

The MathWorks. MATLAB® 7 *Getting Started Guide*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *MATLAB® 7 Graphics*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *MATLAB® 7 Mathematics*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *MATLAB® 7 Programming Fundamentals*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *Simulink® 7 Getting Started Guide*. The MathWorks. Natick, MA. 1990–2009.

The MathWorks. *Simulink® 7 User's Guide*. The MathWorks. Natick, MA. 1990–2009.

Appendix C: MATLAB's Simulink Tutorial

C.1 Introduction

Readers who are studying MATLAB may want to explore the functionality and convenience of MATLAB's Simulink. Before proceeding, the reader should have studied Appendix B, the MATLAB Tutorial, including Section B.1, which is applicable to this appendix.

MATLAB's Simulink Version 7.4(R2009b) and MATLAB Version 7.9(R2009b) are required in order to use Simulink.

The models described in this appendix, which are available at www.wiley.com/college/nise, were developed on a PC using MATLAB Version 7.9 and Simulink Version 7.4. The code will also run on workstations that support MATLAB. Consult the MATLAB Installation Guide for your platform for minimum system hardware requirements.

Simulink is used to simulate systems. It uses a graphical user interface (GUI) for you to interact with blocks that represent subsystems. You can position the blocks, resize the blocks, label the blocks, specify block parameters, and interconnect blocks to form complete systems from which simulations can be run.

Simulink has block libraries from which subsystems, sources (that is, function generators), and sinks (that is, scopes) can be copied. Subsystem blocks are available for representing linear, nonlinear, and discrete systems. LTI objects can be generated if the Control System Toolbox is installed.

Help is available on the menu bar of the **MATLAB** Window. Under **Help** select **Product Help.** When the help screen is available, choose **Simulink** under the **Contents** tab. Help is also available for each block in the block library and is accessed either by right-clicking a block's icon in the **Simulink Library Browser** and selecting **Help for** . . . or by double-clicking the block's icon and then clicking the **Help** button. Finally, screen tips are available for some toolbar buttons. Let your mouse's pointer rest on the button for a few seconds to see the explanation.

C.2 Using Simulink

The following summarize the steps to take to use Simulink. Section C.3 will present four examples that demonstrate and clarify these steps.

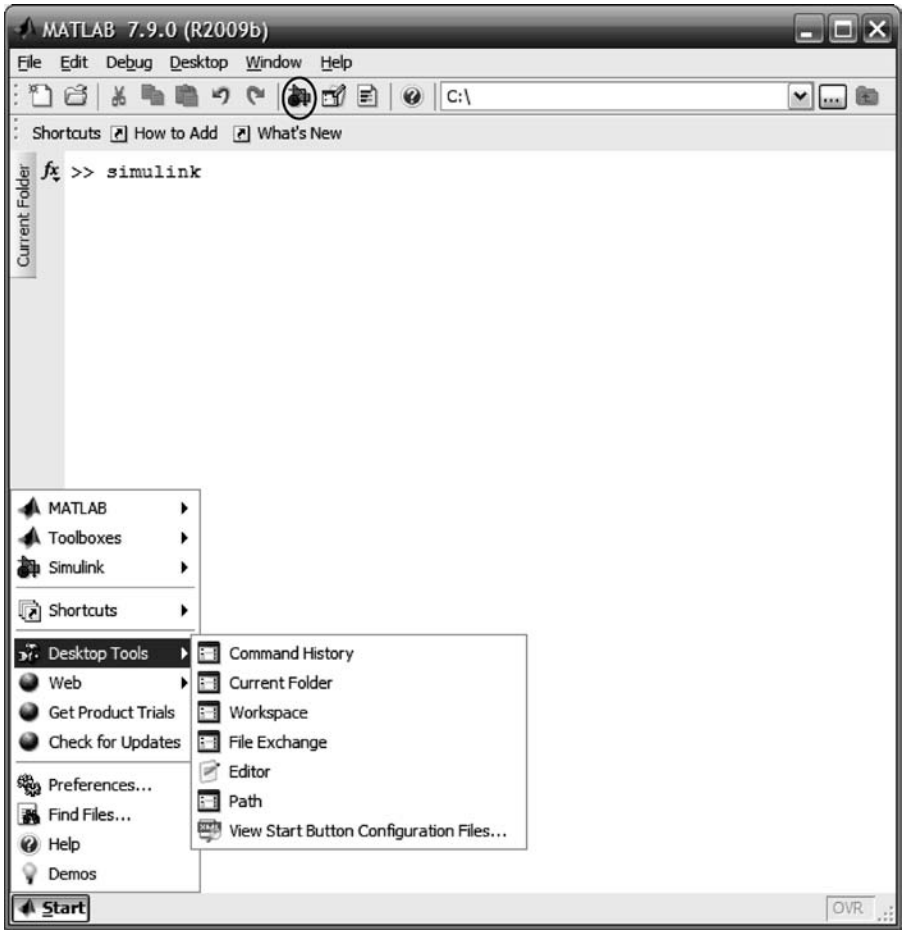


FIGURE C.1 **MATLAB** Window showing how to access Simulink. The **Simulink Library Browser** button is shown circled.

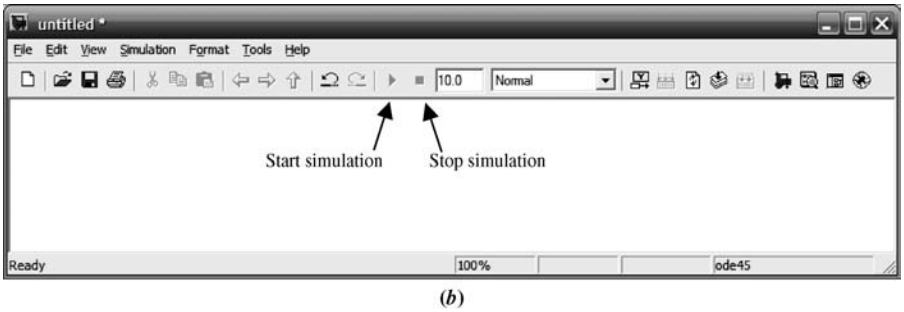
1. **Access Simulink** The **Simulink Library Browser**, from where we begin Simulink, is accessed by typing *simulink* in the **MATLAB Command Window** or by clicking on the **Simulink Library Browser** button on the toolbar, shown circled in Figure C.1.

   In response, MATLAB displays the **Simulink Library Browser** shown in Figure C.2(*a*). We now create an **untitled** window, Figure C.2(*b*), by clicking on the **Create a new model** button (shown circled in Figure C.2(*a*)) on the tool bar of the **Simulink Library Browser**. You will build your system in this window. Existing models may be opened by clicking on the **Open a model** button on the **Simulink Library Browser** toolbar. This button is immediately to the right of the **Create a new model** button. Existing models may also be opened by selecting the **Current Folder** from the **command Window Start** menu or the tab on the left side of the **Command Window as** shown in **Current** Figure C.1, selecting your file names, and then dragging them to the **MATLAB Command Window**.

2. **Select blocks** Figure C.2(*a*) shows the **Simulink Library Browser** from which all blocks can be accessed. The left-hand side of the browser shows major libraries, such as **Simulink**, as well as underlying block libraries, such as **Continuous**. The



(*a*)

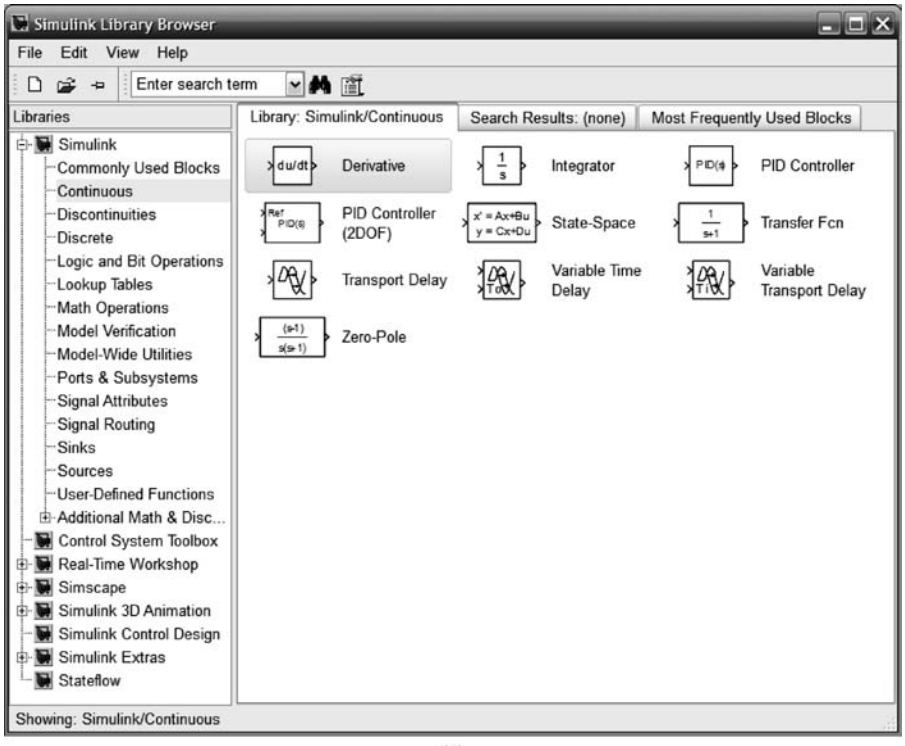


(*b*)

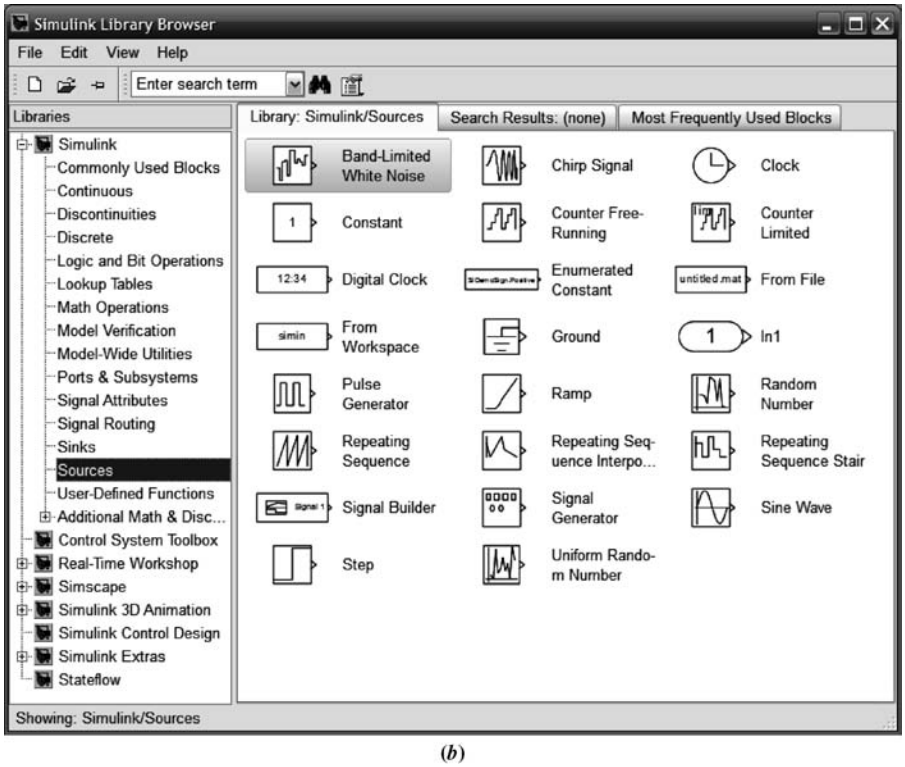
FIGURE C.2 **a. Simulink Library Browser** window showing the **Create a new model** button encircled **b.** resulting **untitled** model window

right-hand side of Figure C.2(*a*) also shows the underlying block libraries. To reveal a block library's underlying blocks, select the block library on the left-hand side or double-click the block library on the right-hand side. As an example, the **Continuous** library blocks under the **Simulink** major library are shown exposed in Figure C.3(*a*). Figures C.3(*b*) and C.3(*c*) show some of the **Sources** and **Sinks** library blocks, respectively.

Another approach to revealing the Simulink block library is to type *open_system ('simulink.mdl')* in the **MATLAB Command Window**. The window shown in Figure C.4 is the result. Double-clicking any of the libraries in Figure C.4

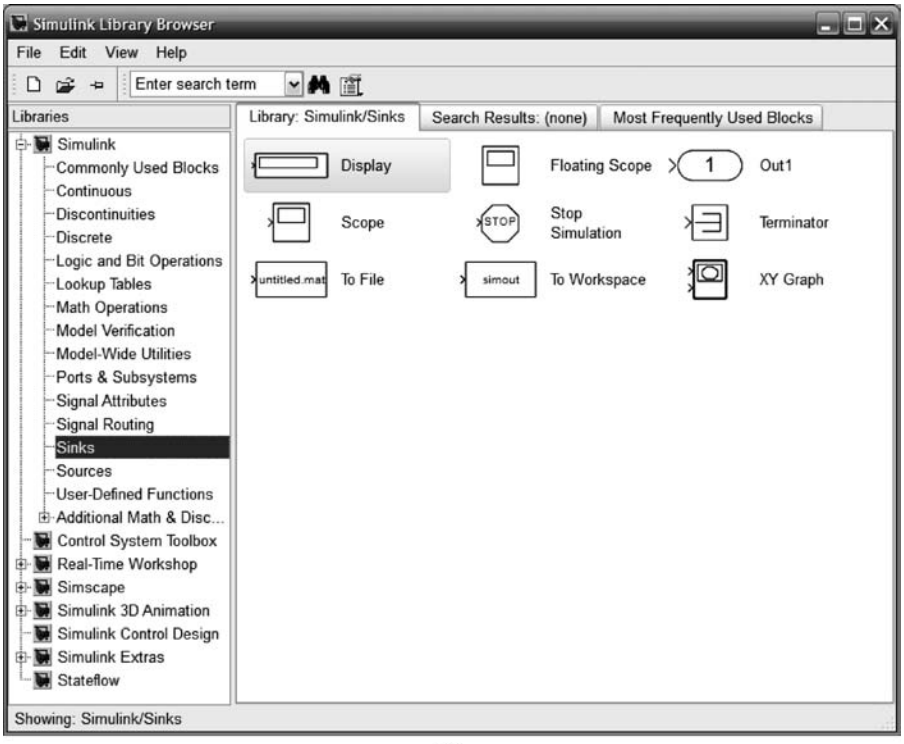


(*a*)



(*b*)

FIGURE C.3 Simulink block libraries: **a.** Continuous systems **b.** Sources (figure continues)



(*c*)

FIGURE C.3 (continued) **c.** Sinks

reveals an individual window containing that library's blocks, equivalent to the right-hand side of the **Simulink Library Browser** as shown in the examples of Figure C.3.

3. **Assemble and label subsystems** Drag required subsystems (blocks) to your model window from the browser, such as those shown in Figure C.3. Also, you may access the blocks by double-clicking the libraries shown in Figure C.4. You can position, resize, and rename the blocks. To position, drag with the mouse; to resize, click on the subsystem and drag the handles; to rename, click on the existing name, select the existing text, and type the new name. The text can also be repositioned to the top of the block by holding the mouse down and dragging the text.
4. **Interconnect subsystems and label signals** Position the pointer on the small arrow on the side of a subsystem, press the mouse button, and drag the resulting cross-hair pointer to the small arrow of the next subsystem. A line will be drawn between the two subsystems. Blocks may also be interconnected by single-clicking the first block followed by single-clicking the second block while holding down the control key. You can move line segments by positioning the pointer on the line, pressing the mouse button, and dragging the resulting four-arrow pointer. Branches to line segments can be drawn by positioning the pointer where you want to create a line segment, holding down the mouse's right button, and dragging the resulting cross hairs. A new line segment will form. Signals can be labeled by double-clicking the line and typing into the resulting box. Finally, labels can be placed anywhere by double-clicking and typing into the resulting box.

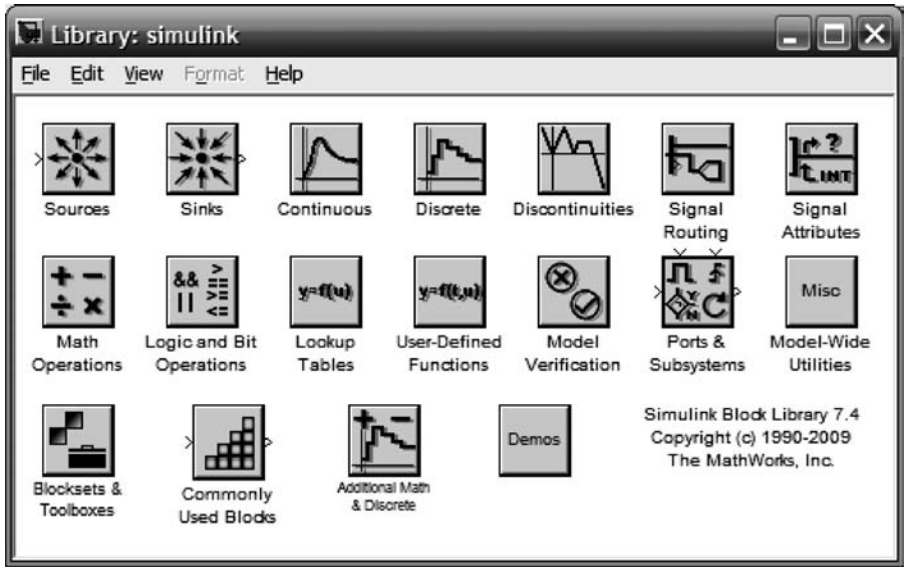


FIGURE C.4 Simulink Block Library window

5. **Choose parameters for the subsystems** Double-click a subsystem in your model window and type in the desired parameters. Some explanations are provided in the **Parameters** window. Press the Help button in the **Parameters** window for more details. The parameters can be read later without opening the block. Let your mouse's pointer rest on the block for a few seconds, and a screen tip will appear, identifying the block and listing its parameters. The information displayed in the screen tip first must be selected in the **Block Data Tips Options** in the model window's **View** menu. Explore other options by right-clicking on a block.
6. **Choose parameters for the simulation** Select **Configuration parameters** . . . under the **Simulation** menu in your model window to set additional parameters, such as simulation time. Press the **Help** button in the **Configuration parameters** window for more details.
7. **Start the simulation** Make your model window the active window. Double-click the **Scope** block (typically, the scope is used to view the simulation results) to display the **Scope** window. Select **Start** under the **Simulation** menu in your model window or click on the **Start simulation** icon on the toolbar of your model window as shown in Figure C.2(*b*). Clicking the **Stop simulation** icon will stop the simulation before completion.
8. **Interact with the plot** In the **Scope** window, using the toolbar buttons, you can zoom in and out, change axes ranges, save axis settings, and print the plot. Right-clicking on the **Scope** window brings up other choices.
9. **Save your model** Saving your model, by choosing **Save** under the **File** menu, creates a file with an .mdl extension, which is required.

C.3 Examples

This section will present four examples of the use of Simulink to simulate linear, nonlinear, and digital systems. Examples will show the Simulink block diagrams as well as explain the settings of parameters for the blocks. Finally, the results of the simulations will be shown.

Example C.1

Simulation of Linear Systems

Our first example develops a simulation of three linear systems to compare their step responses. In particular, we solve Example 4.8 and reproduce the responses shown in Figure 4.24. Figure C.5 shows a Simulink block diagram formed by following Steps (1) through (5) in Section C.2 as follows:

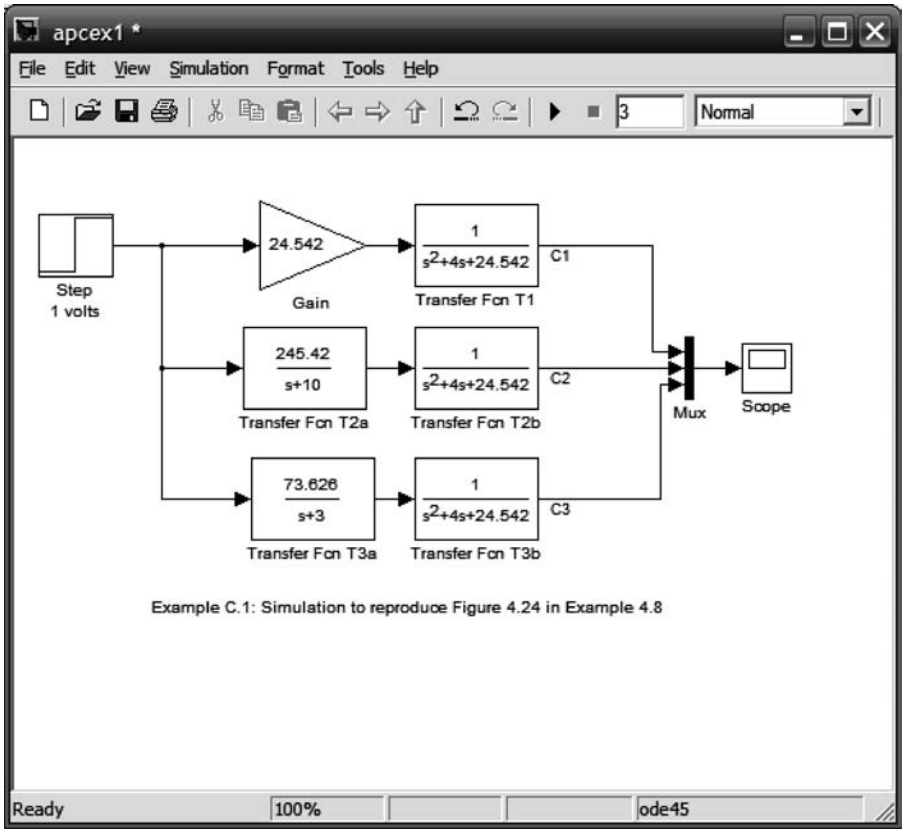


FIGURE C.5 Simulink block diagram for Example C.1

Access Simulink; select, assemble, and label subsystems The source is a 1-volt step input, obtained by dragging the **Step** block from the **Simulink Library Browser** under **Sources** to your model window.

The first system, T1, consists of two blocks, **Gain** and **Transfer Fcn**. Gain is obtained by dragging the **Gain** block from the **Simulink Library Browser** under **Math Operations** to your model window. Transfer function, T1, is obtained by dragging the **Transfer Fcn** block from the **Simulink Library Browser** under **Continuous** to your model window. Systems T2 and T3 are created similarly.

The three output signals, C1, C2, and C3, are multiplexed for display into the single input of a scope. The Mux (multiplexer) is obtained by dragging the **Mux** block from the **Simulink Library Browser** under **Signal Routing** to your model window.

The sink is a scope, obtained by dragging the **Scope** block from the **Simulink Library Browser** under **Sinks** to your model window.

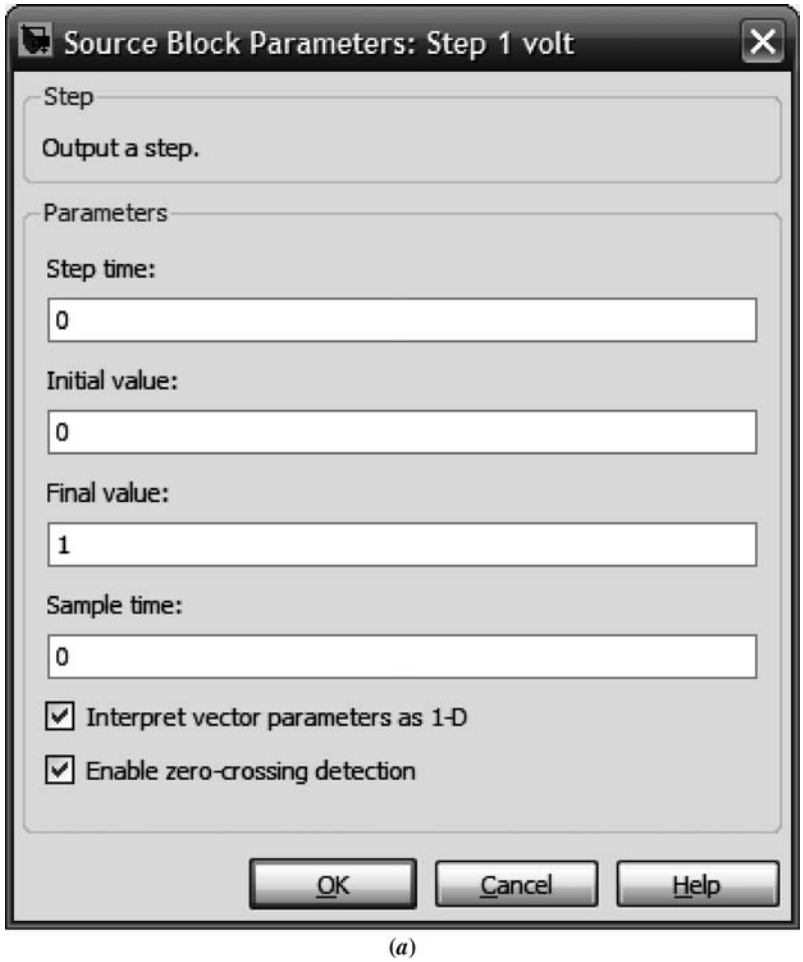
Alternatively, all blocks can be dragged from the **Library: simulink** window shown in Figure C.4. The **Mux** can be found under **Signal Routing** in the **Library: simulink** window.

The labels for the blocks can be changed to those shown in Figure C.5 by following Step (3) in Section C.2.

Interconnect subsystems and label signals Follow Step (4) to interconnect the subsystems and label the signals. You must set the mux's parameters before the wiring can be completed. See the next paragraph.

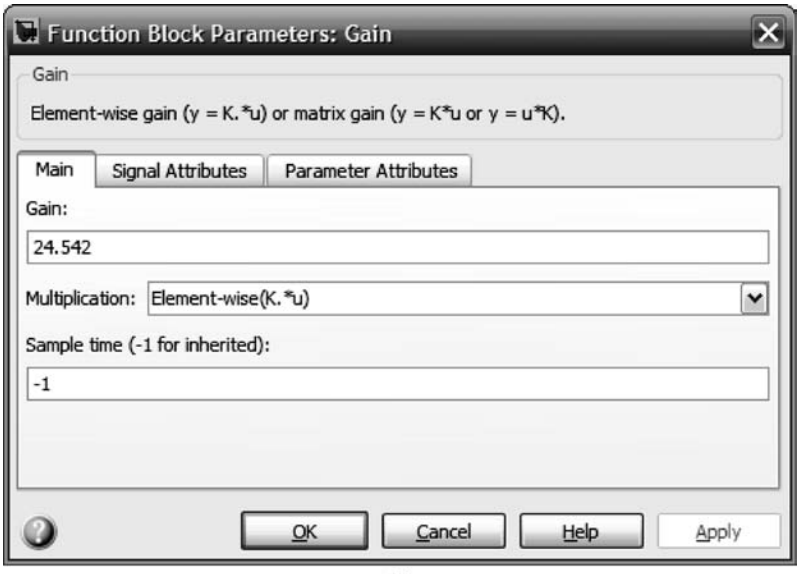
Choose parameters for the subsystems Let us now set the parameters of each block using Step (5). The **Block Parameters** window for each block is accessed by double-clicking the block on your model window. Figure C.6 shows the **Block Parameters** windows for the 1 volt step input, gain, transfer function 1, and mux. Set the parameters to the required values as shown.

The scope requires further explanation. Double-clicking the **Scope** block in your model window accesses the scope's display, Figure C.7(*a*).

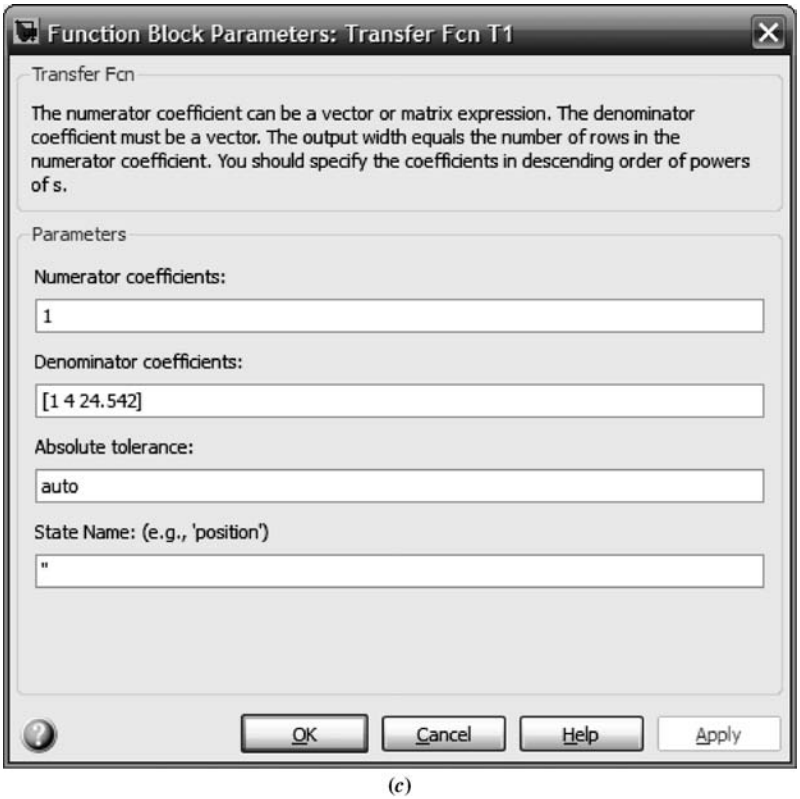


(*a*)

FIGURE C.6 **Block parameters** windows for **a**. 1 volt step source; (figure continues)

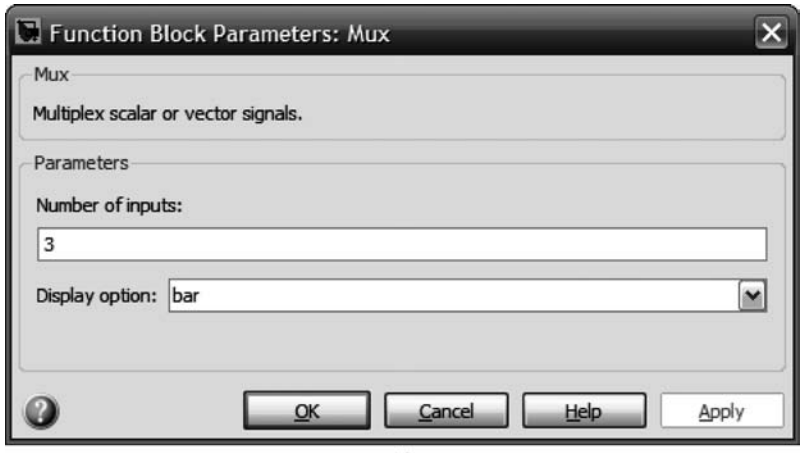


(*b*)



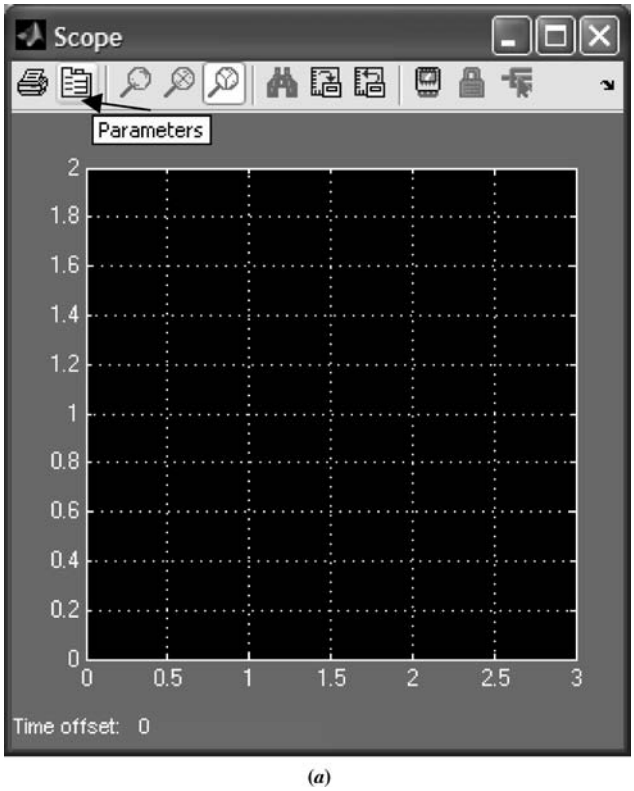
(*c*)

FIGURE C.6 **b**. gain; **c**. transfer function 1; (figure continues)

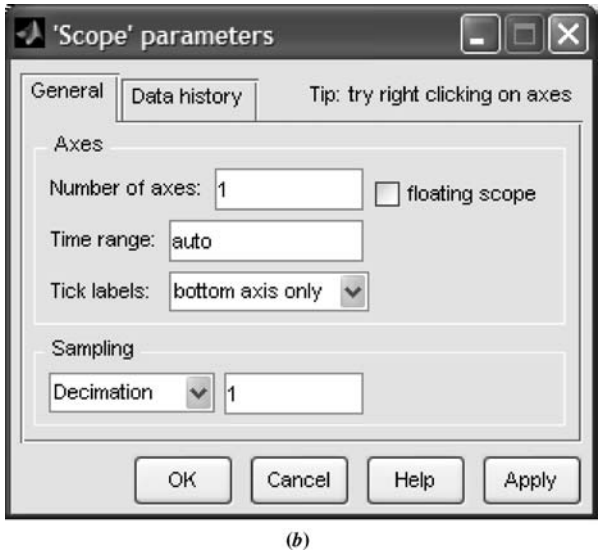


(*d*)

FIGURE C.6 (continued) **d**. mux

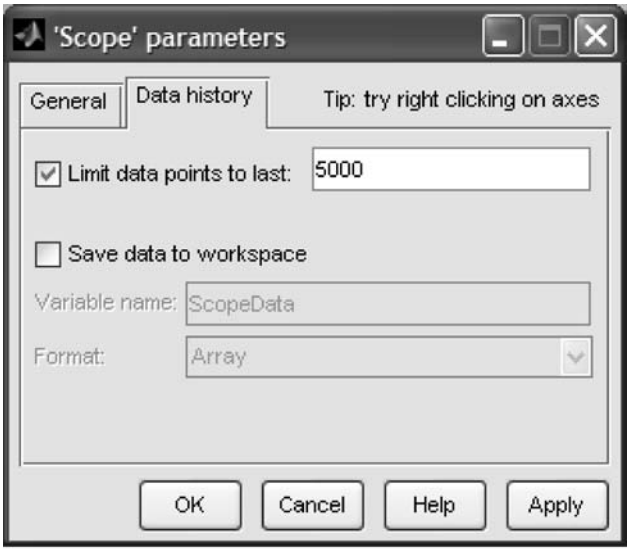


(*a*)

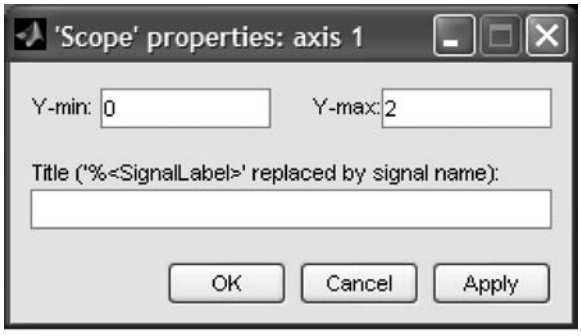


(*b*)

FIGURE C.7 Windows for the scope: **a. Scope; b. 'Scope' parameters, General** tab; (figure continues)



(*c*)



(*d*)

FIGURE C.7 (continued) **c. 'Scope' parameters, Data history** tab; **d. 'Scope' properties: axis 1**

Clicking the **Parameters** icon on the **Scope** window toolbar, shown in Figure C.7(*a*), accesses the **'Scope' parameters** window as shown in Figure C.7(*b*). The **'Scope' parameters** window contains two tabs, **General** and **Data history**, as shown in Figure C.7(*b*) and (*c*), respectively. Finally, right-clicking in the plotting area in the Scope window and selecting **Axis properties** . . . reveals the **'Scope' properties: axis 1** window, Figure C.7(*d*). We now can set the display parameters, such as amplitude range.

Choose parameters for the simulation Follow Step (6) to set simulation parameters. Figure C.8 shows the resulting **Configuration Parameters** window after

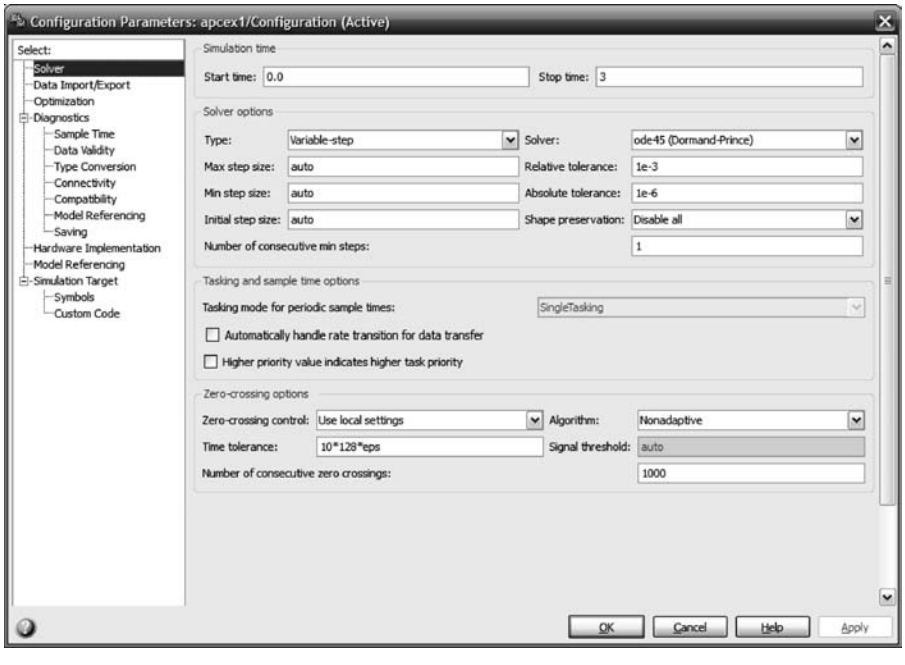


FIGURE C.8 **Simulation Parameters** window for **Solver tab**

selecting the **Solver** tab. Among other parameters, the simulation start and stop times can be set.

Start the simulation Now run the simulation by following Step (7). Figure C.9 shows the result in the **Scope** window. Plots are color coded in the order in which they appear at the mux input as follows: yellow, magenta, cyan, red, green, and dark blue. If the mux has more inputs, the colors recycle.

Interact with the plot The toolbar of the **Scope** window shown in Figure C.9 has several buttons that can be used to interact with the plot. Let us summarize the function and operation of each, starting with the left-most button:

Button 1 executes a plot print.
Button 2 has already been explained and is used to set scope parameters.
Button 3 permits zooming into the plot in both the x and y directions. Press the button and drag a rectangle over the portion of the curve you want to expand.
Button 4 allows zooming in the x direction only. Drag a horizontal line over the plot covering the extent of x you want to expand.
Button 5 allows zooming in the y direction only. Drag a vertical line over the plot covering the range of y you want to expand.
Button 6 autoscales axis for use after zooming.
Button 7 saves current axis settings.
Button 8 restores saved axis settings.

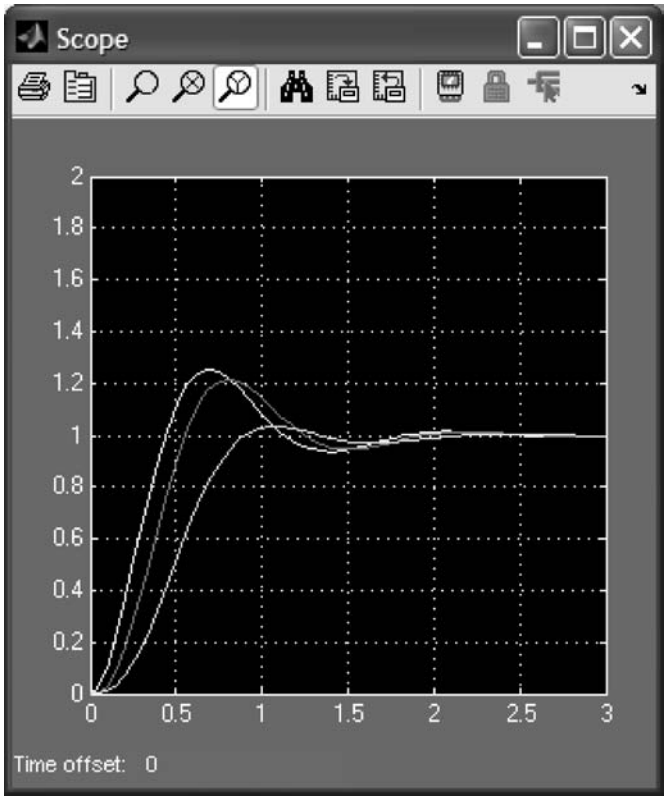


FIGURE C.9 **Scope** window after Example C.1 simulation stops

Button 9 toggles floating scope. It must be turned off to use zooming. See documentation for use of floating scopes.
Button 10 toggles lock for current axis selection.
Button 11 allows selection of signals to view when using floating scope.

Example C.2

Effect of Amplifier Saturation on Motor's Load Angular Velocity

This example, which generated Figure 4.29 in the text, shows the use of Simulink to simulate the effect of saturation nonlinearity on an open-loop system. Figure C.10 shows a Simulink block diagram formed by following Steps (1) through (5) in Section C.2 above.

Saturation nonlinearity is an additional block that we have not used before. Saturation is obtained by dragging to your model window the **Saturation** block in the **Simulink Library Browser** window under **Discontinuities** as shown in Figure C.11(*a*) and setting its parameters to those shown in Figure C.11(*b*).

Now run the simulation by making your model window active and selecting **Start** under the **Simulation** menu of your model window or clicking on the **Start simulation** button on your model window toolbar. Figure C.12 shows the result in the **Scope** window.

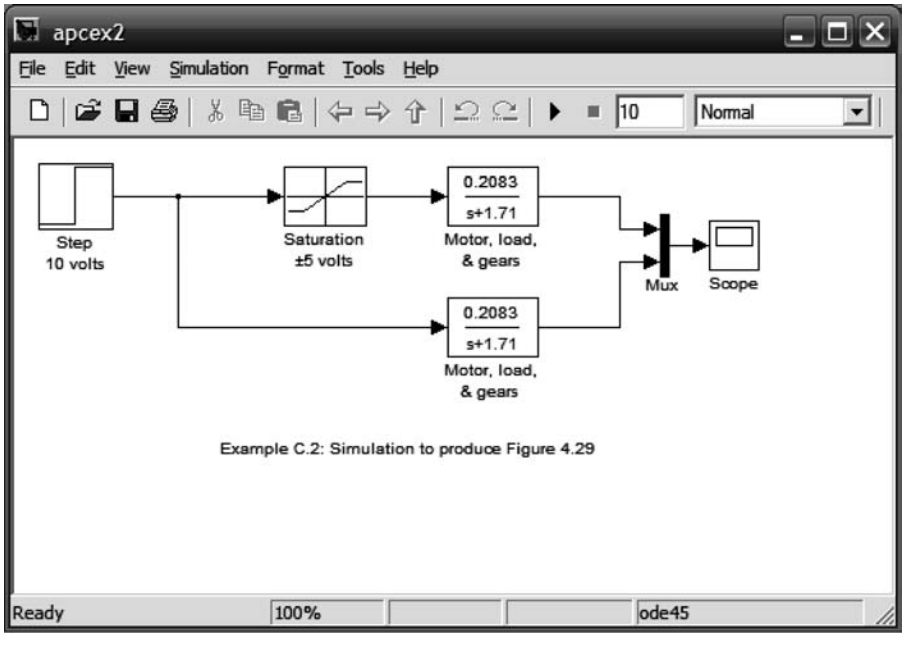
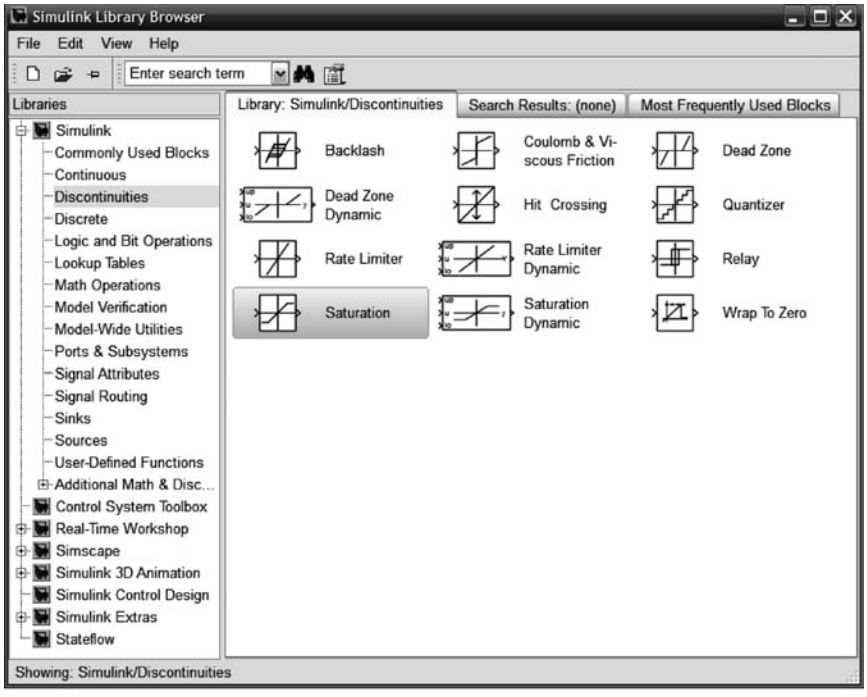
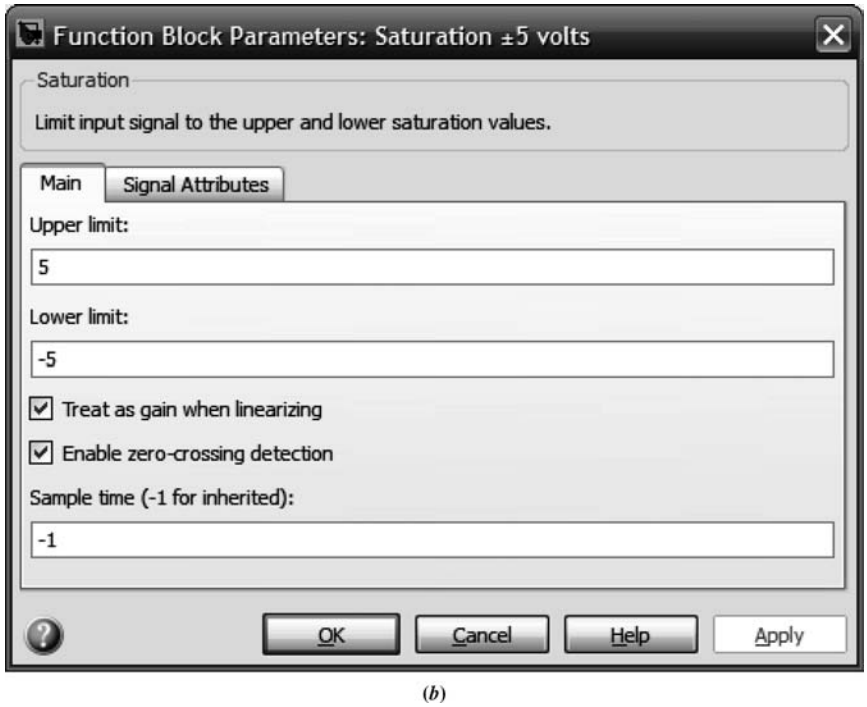


FIGURE C.10 Simulink block diagram for Example C.2



(*a*)



(*b*)

FIGURE C.11 **a.** Simulink library for nonlinearities; **b.** parameter settings for saturation

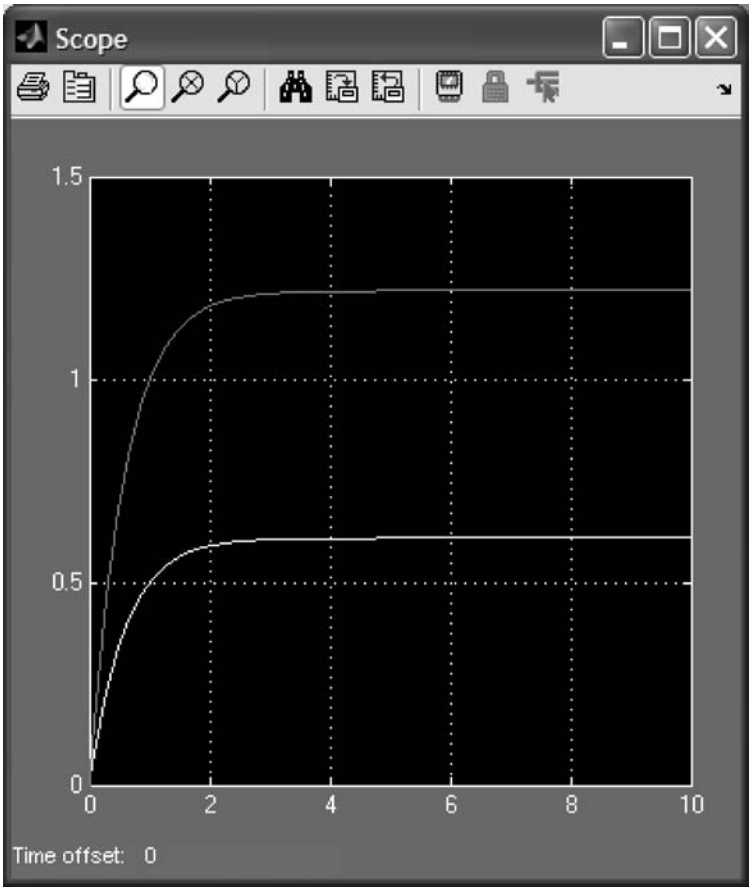


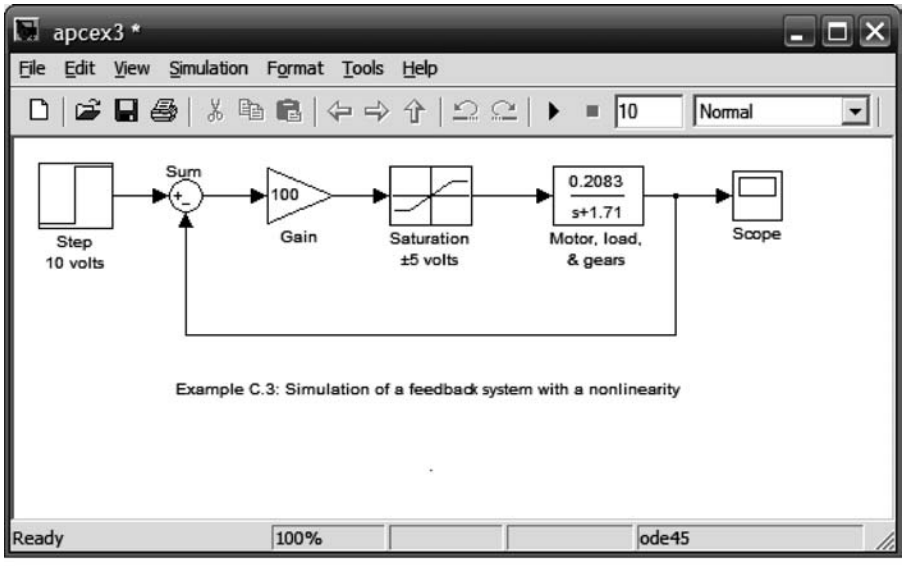
FIGURE C.12 **Scope** window after simulation of Example C.2 stops. The lower curve is the output with saturation

Example C.3

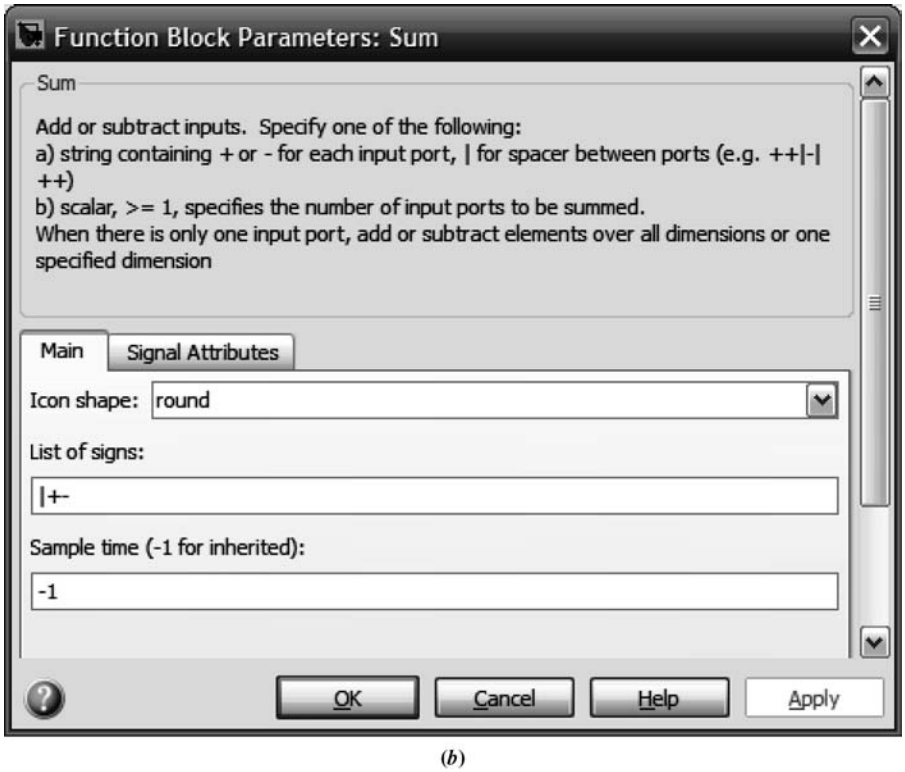
Simulating Feedback Systems

Simulink can be used for the simulation of feedback systems. Figure C.13(*a*) is an example of a feedback system with saturation.

In this example, we have added a feedback path (see Step (4) in Section C.2) and a summing junction, which is obtained by dragging the **Sum** block from the **Simulink Library Browser**, contained in the **Math Operations** library, to your model window. The **Function Block Parameters: Sum** window, Figure C.13(*b*), shows the parameter settings for the summer. You can set the shape as well as set the plus and minus inputs. In the list of signs, the "|"symbol signifies a space. We place it at the beginning to start the signs at "nine o'clock,'' conforming to our standard symbol, rather than at "12 o'clock." The result of the simulation is shown in Figure C.14.



(*a*)



(*b*)

FIGURE C.13 **a.** Simulation block diagram for a feedback system with saturation; **b.** block parameter window for the summer

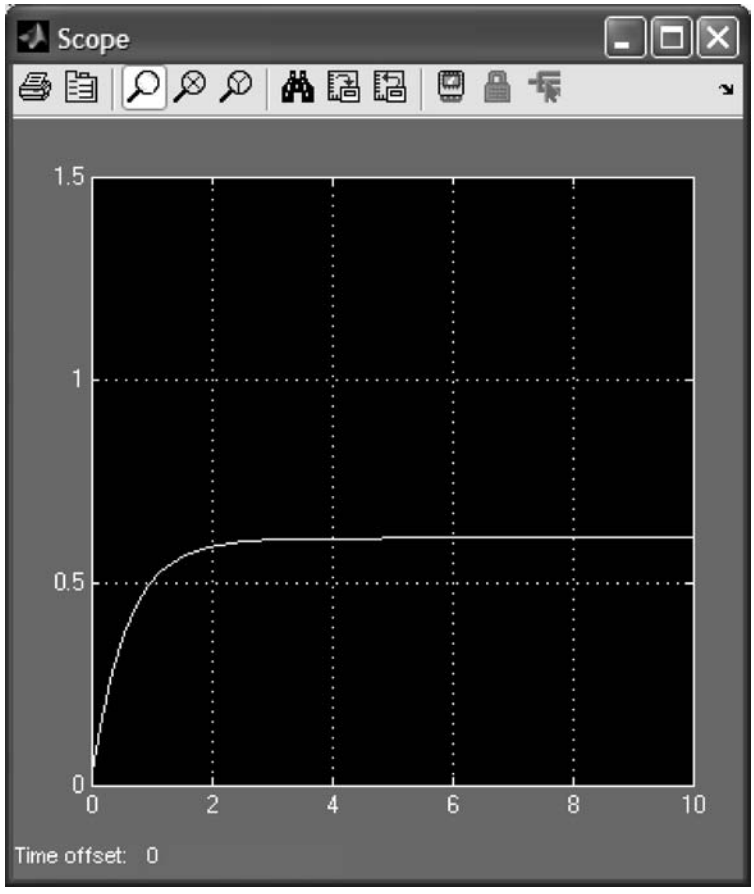


FIGURE C.14 Simulation output for Example C.3

Example C.4

Simulating Digital Systems

This example demonstrates two methods of generating digital systems via Simulink for the purpose of simulation, as shown in Figure C.15.

The first approach uses a linear transfer function cascaded with a **Zero-Order Hold** block obtained from the **Simulink Library Browser** under the **Discrete** block library, shown on the right-hand side of Figure C.16. The second method uses a discrete transfer function also obtained from the **Simulink Library Browser** under the **Discrete** block library. The remainder of the block diagram was obtained by methods previously described.

The block parameters for the **Zero-Order Hold** and **Discrete Transfer Fcn** blocks are set as shown in Figures C.17(*a*) and (*b*), respectively.

Select **Configuration parameters** . . . under the **Simulation** menu in your model window and set the simulation stop time to 4 seconds, the type **to fixed-step**, and the solver to **ode4 (Runge-Kutta)**. The result of the simulation is shown in Figure C.18.

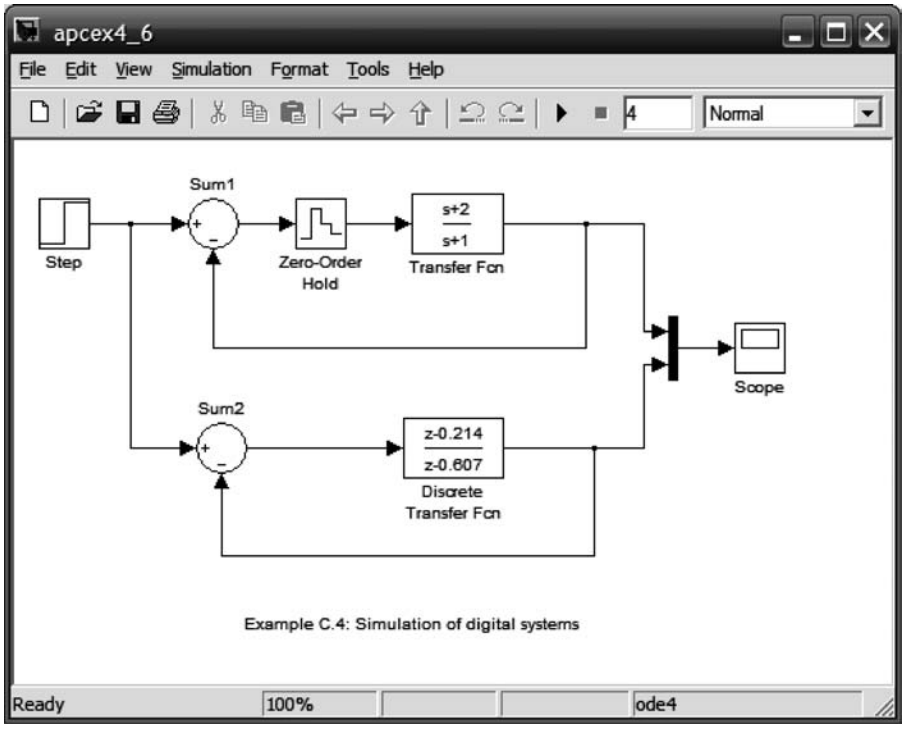


FIGURE C.15 Simulink block diagram for simulating digital systems two ways

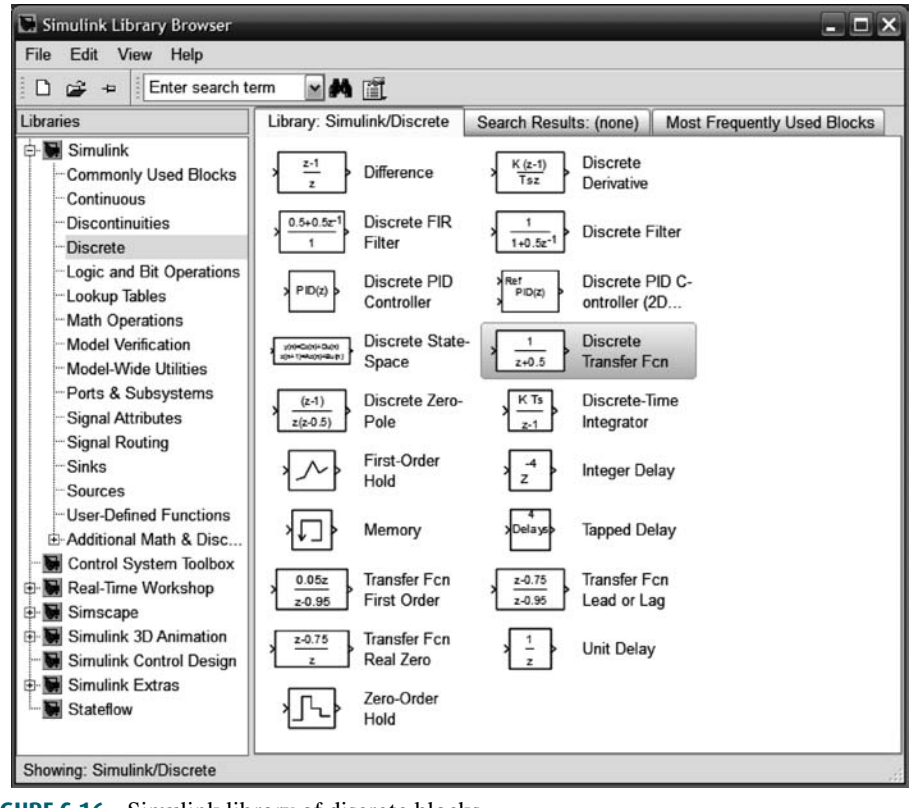
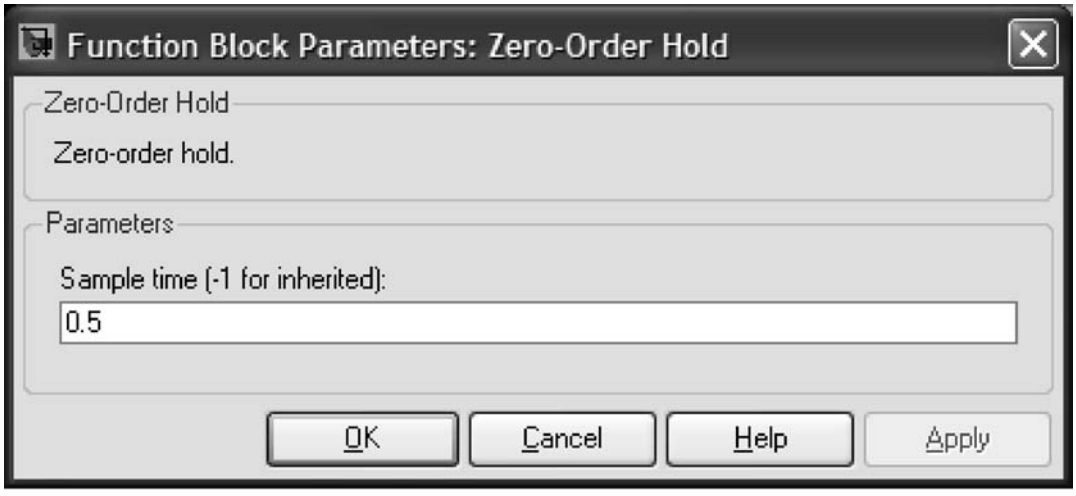
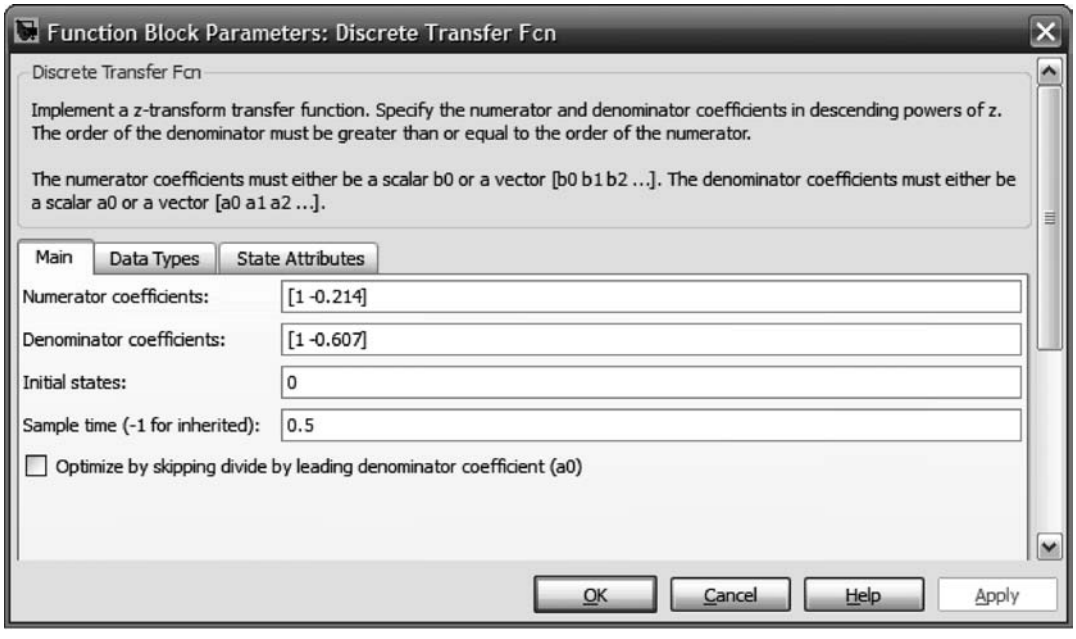


FIGURE C.16 Simulink library of discrete blocks



(*a*)



(*b*)

FIGURE C.17 Function Block parameter windows for: **a. Zero-Order Hold** block; **b. Discrete Transfer Fcn** block

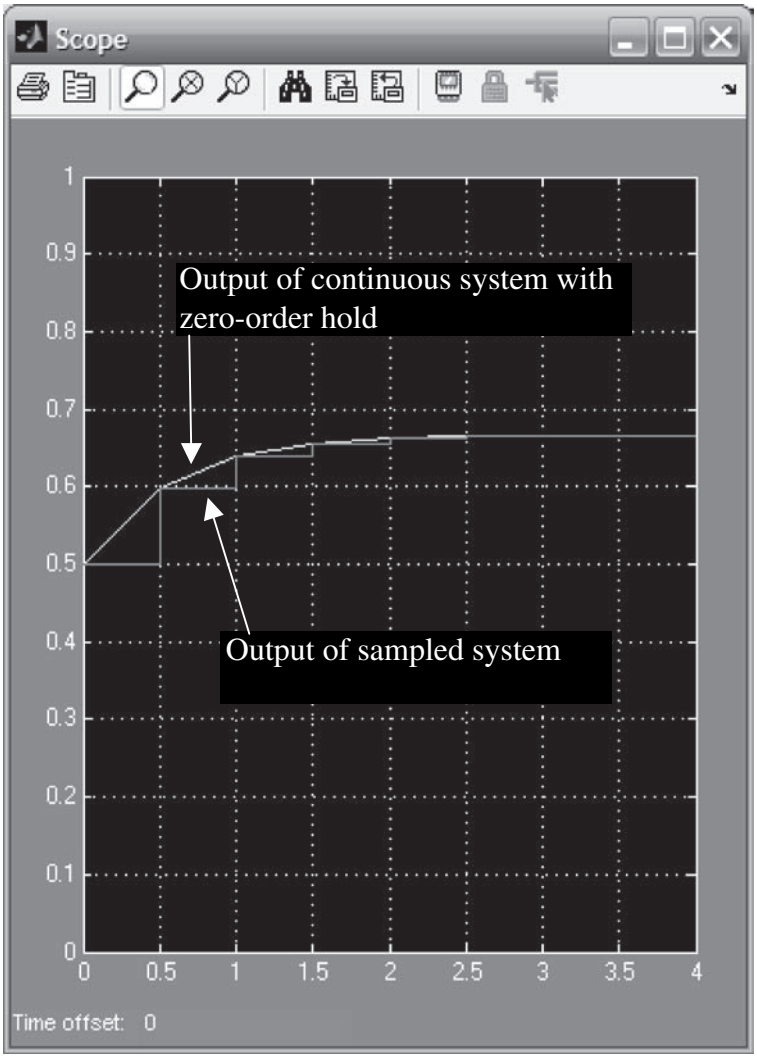


FIGURE C.18 Outputs of the digital systems

Summary

This appendix explained Simulink, its advantages, and how to use it. Examples were taken from Chapters 4, 5, and 13 and demonstrated the use of Simulink for simulating linear, nonlinear, and digital systems.

The objective of this appendix was to familiarize you with the subject and get you started using Simulink. There are many blocks, parameters, and preferences that could not be covered in this short appendix. You are encouraged to explore and expand your use of Simulinkby using the on-screen help that was explained earlier. The references in the Bibliography of this appendix also provide an opportunity to learn more about Simulink.

Bibliography

The MathWorks. *Control System Toolbox ™ 8 Getting Started Guide*. The MathWorks. Natick, MA. 2000–2009.

The MathWorks. *Control System Toolbox ™ 8 User's Guide*. The MathWorks. Natick, MA. 2001–2009.

The MathWorks. *MATLAB® 7 Getting Started Guide*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *MATLAB® 7 Graphics*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *MATLAB® 7 Graphics Programming Fundamentals*. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. *Simulink® 7 Getting Started Guide*. The MathWorks. Natick, MA. 1990–2009.

The MathWorks. *Simulink® 7 User's Guide*. The MathWorks. Natick, MA. 1990–2009.

This page intentionally left blank

Appendix D: LabVIEW™[1] Tutorial

D.1 Introduction

LabVIEW is a programming environment that is presented here as an alternative to MATLAB. Although not necessary, the reader is encouraged to become acquainted with MATLAB before proceeding, since familiarity with MATLAB can enhance the understanding of the relationship between textual (MATLAB) and graphical (LabVIEW) programming languages and extend the functionality of LabVIEW. In this tutorial, we will show how to use LabVIEW to **(1)** analyze and design control systems, and **(2)** simulate control systems. This appendix was developed using LabVIEW 2009.

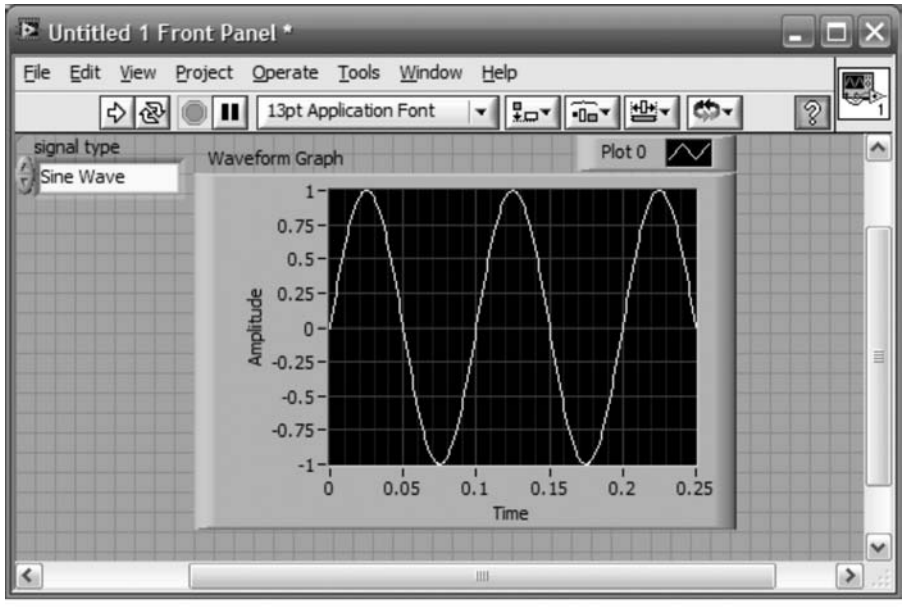
LabVIEW is a graphical programming environment that produces virtual instruments (VI's). A VI is a pictorial reproduction of a hardware instrument on your computer screen, such as an oscilloscope or waveform generator. The VI can consist of various controls and indicators, which become inputs and outputs, respectively, to your program. Underlying each control and indicator is an associated block of code that defines its operation. The LabVIEW model thus consists of two windows: **(1) Front Panel**, which is a replica of the hardware front panel showing the controls and indicators, and **(2) Block Diagram**, which contains the underlying code for the controls and indicators on the **Front Panel**.

Associated with the **Front Panel** window is a **Controls** palette window containing numerous icons representing controls and indicators. The icons can be dragged onto a **Front Panel** window to create that control or indicator. Simultaneously, the associated code block is formed on the **Block Diagram** window.

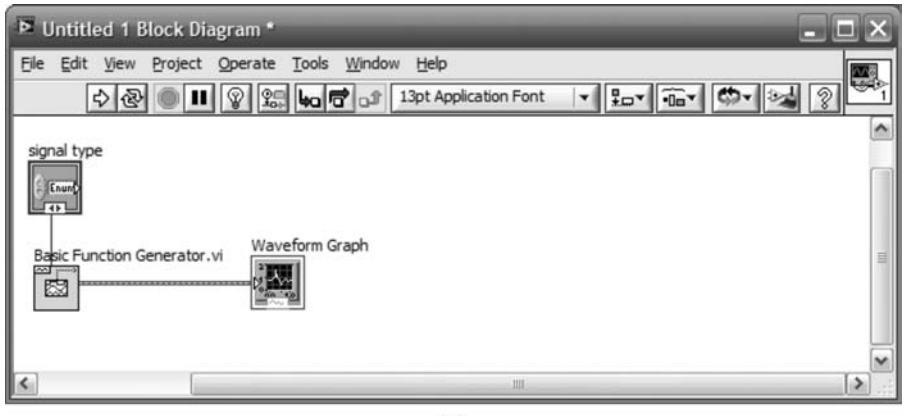
Alternately, the block diagram can be formed first, and then the front panel is created from the block diagram. Associated with the **Block Diagram** window is a **Functions** palette window containing numerous icons representing a wide range of functions. Icons can be dragged onto a **Block Diagram** window to create that code block.

For example, Figure D.1(*a*) is the front panel of a signal generator. The generator consists of a control to select the signal type and a waveform graph that shows the output waveform. Figure D.1(*b*) shows the underlying code, which is contained in the code blocks. Here, the signal type selector is a control, while the waveform graph is an indicator. Later we will show how to make connections to other VI's. The palette windows for the front panel and block diagram are shown respectively in Figures D.1(*c*) and (*d*).

[1] LabVIEW is a registered trademark of National Instruments Corporation.



(*a*)



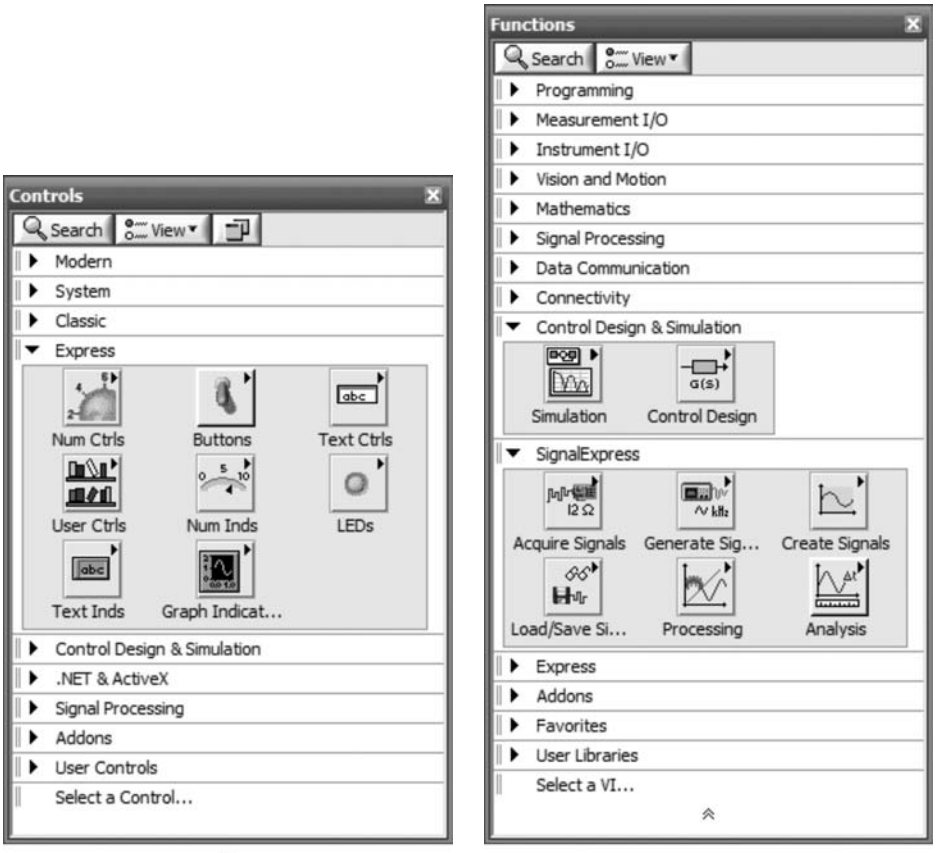
(*b*)

FIGURE D.1 A LabVIEW function generator VI: **a. Front Panel** window; **b. Block Diagram** window; (*figure continues*)

D.2 Control Systems Analysis, Design, and Simulation

LabVIEW can be used as an alternative to or in conjunction with MATLAB to analyze, design, simulate, build, and deploy control systems. In addition to LabVIEW, you will need the LabVIEW Control Design and Simulation Module. Finally, as an option that will be explained later, you may want to install the MathScript RT Module.

Analysis and design can be thought of as similar to writing MATLAB code, while simulation can be thought of as similar to Simulink. In LabVIEW, analysis and design, as opposed to simulation, are handled from different subpalettes of the **Functions** window's **Control Design and Simulation** palette. See Figure D.1(*d*). Analysis and design, and simulation will typically begin with the **Block Diagram** window, where icons representing code blocks will be interconnected. Parameters



(*c*) (*d*)

FIGURE D.1 (*Continued*) **c. Controls** palette; **d. Functions** palette

used by the code can be conveniently selected, changed, and passed to the code through VI controls on the **Front Panel** window created from the code icons. Any results, such as time response, can be displayed through VI indicators on the **Front Panel** window created from the code icons.

D.3 Using LabVIEW

The following steps start you on your way to using LabVIEW for control systems analysis, design, and simulation. These steps will be illustrated in the examples that follow.

1. **Start LabVIEW** LabVIEW starts with the **Getting Started** window shown in Figure D.2, where you can select a **New** file or **Open** an existing file. You may also select various resources. Selecting **Blank VI** under the **New** label or **New VI** under the **File** menu brings up the **Front Panel** and **Block Diagram** windows shown in Figure D.1. If necessary, a window can be opened from the **Window** tab on the menu bar of the **Front Panel** and **Block Diagram**.

   Right-click the **Block Diagram** window to bring up the **Functions** palette and click the thumb tack in the upper left-hand corner to dock the window. Repeat for the **Front Panel** window to access the **Controls** palette.

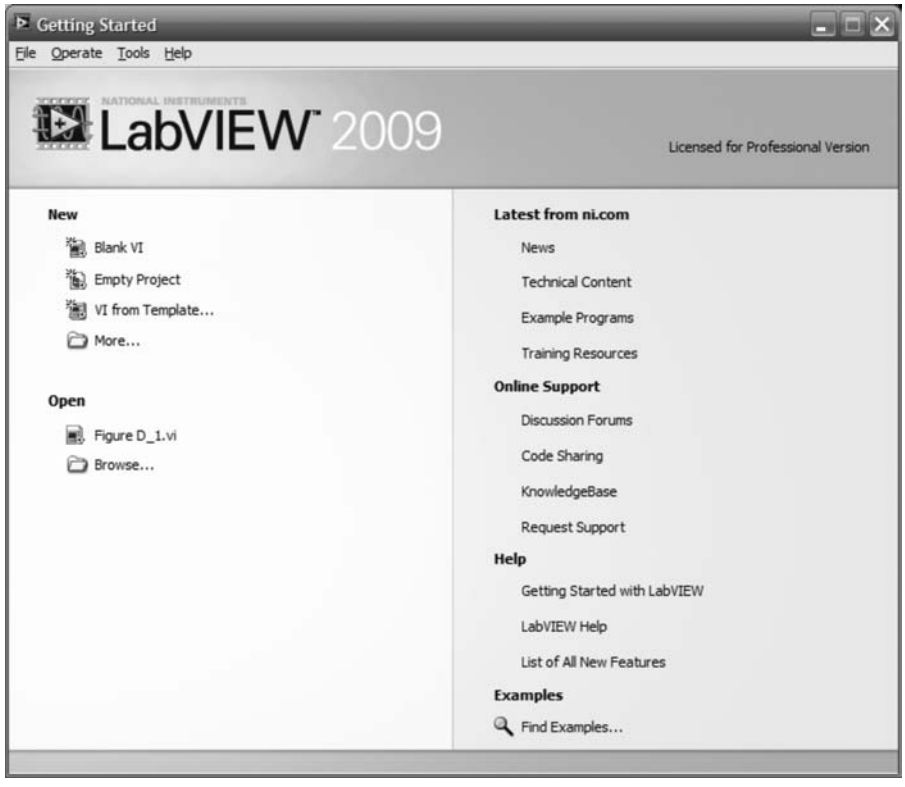


FIGURE D.2 LabVIEW's **Getting Started** window

2. **Select blocks** Make the **Block Diagram** window active, or access it from **Window** on the menu bar. Right-click the **Block Diagram** window or use the **View** menu to bring up the **Functions** palette. Expand the palette window by clicking the double-up arrows at the bottom. At the top of the palette window click **View**, and select **View This Palette As/Category (Icons and Text)** to add a text description below each icon. For control systems analysis, design, and simulation, expand **Control Design & Simulation** in the **Functions** palette by clicking the arrow to the left of this category.

   If you are performing a simulation click the subpalette **Simulation**. If you are performing control system analysis or design, click the subpalette **Control Design**. An arrow in the upper-right corner of a subpalette indicates additional underlying palettes or blocks.

   If the name of the icon is incomplete, resting the mouse over the icon will bring up its complete identification. To obtain detailed help about an icon, right-click the icon and select **Help**.

3. **Move blocks to the block diagram window** To move the icon to the **Block Diagram**, left-click the mouse to attach the icon (some icons take a little time to complete this operation). When the pointer turns into a hand, click the spot on the **Block Diagram** where you want to place the icon.

4. **Obtain information about the block** You will now want to obtain information about how to interconnect the block to other blocks and pass parameters to the block as well as other characteristics about the block. Select the yellow question mark at the right of the **Block Diagram** toolbar to turn on the **Context Help window**. This window will provide help about a particular icon if you rest your

mouse over that icon. Additional help is available under the **Help** menu on the **Block Diagram** menu bar. Finally, right-click the icon to bring up a menu with additional choices, such as **Properties**, if any. In particular, you will use this menu to create the block's front panel's controls and indicators. This front panel will be your interface with the block to choose parameters and see responses.

5. **Interconnect and label blocks** Once blocks are placed on the **Block Diagram** they can be moved about by clicking on them or dragging your mouse across several or them to establish a selection pattern. After the selection pattern has been established, depress the mouse left button and drag to a new location. To delete a block, select the block and press the Delete button.

   The context help for the block includes a description of the block's terminals. Let your mouse rest on a terminal until the mouse pointer turns into a spool of wire. Click the terminal and then move the mouse to the next icon's terminal where you want to make the connection. Click the destination terminal to complete the wiring. Notice that the terminal in the **Context Help** window blinks when your mouse resides above that terminal on the block, ensuring that you are on the correct terminal. If you make an error in wiring, click on the wire and press the Delete button or right-click the wire and select **Delete Wire Branch**.

   Block labels can be displayed or hidden. Right-click on the block to bring up the pop-up menu and check or uncheck **Visible Items/Label** to display or hide, respectively, the label. Double-clicking on the label above some blocks will allow you to select and change the text. One click of the mouse on the label will place a selection pattern around the label and allow you to hold down the left key of the mouse and move the label to a different location.

6. **Create the interface to your block** You will now want to create the interface to your block in order to control or select functions, specify parameters, or view responses. This interface will be accessed via the **Front Panel** window. Right-click a terminal on a block for which you want to create an interface. On the pop-up menu, choose **Create/Control** to be able to interact with the block or **Create/Indicator** to view a response or setting.

7. **Set the controls** Switch to the **Front Panel** window and set your controls. For example, enter parameter values, select functions, etc. If you want to change values and at some future time return to the current values, click on **Edit** on the **Block Diagram** menu bar and select **Make Current Values Default**. To return to the default values in the future, click on **Edit** on the **Block Diagram** menu bar and select **Reinitialize Values to Default**.

8. **Run the program** Click on the arrow at the left of the toolbar on either the **Block diagram** or **Front Panel** window to run the program. The program can be run continuously by clicking the curved arrow button on the toolbar second from the left. Continuously running your program permits changing functions and parameter values during execution.

   In order to identify the buttons, let your mouse rest on a button to bring up a context menu. Stop your simulation by pressing the red-dot button, third from the left. If you are performing control systems analysis and design, another way to continuously run the program is to place a **While Loop** around your block diagram. The loop is available in the **Functions** palette at **Express/Execution Control/While Loop**. This loop also places a **Stop** button on the **Front Panel**. The program executes until you press the stop button. In lieu of the **Stop** button, any true/false Boolean can be wired to the condition block (red dot) created inside the **While Loop**.

If you are performing simulation, you can use a **Simulation Loop** available in the **Functions** palette at **Control Design and Simulation/Simulation/Simulation Loop**. Place the **Simulation Loop** around your simulation block diagram by dragging the mouse. Right-click on the **Simulation Loop** outline and choose **Configure Simulation Parameters** . . . to determine the parameters for executing the simulation. The **Front Panel** indicators and controls are also configurable. Right-click on the indicator or control and select **Properties**.

D.4 Analysis and Design Examples

In this section, we will present some examples showing the use of LabVIEW for the analysis and design of control systems. In the next section, examples of the use of LabVIEW for simulation will be presented.

Analysis and design examples use icons selected from the **Control Design** subpalette under the **Control Design and Simulation** palette. In the next section showing examples of simulation, we will use icons taken from the **Simulation** subpalette under the **Control Design and Simulation** palette.

Example D.1

Open-Loop Step Response

TryIt D.1

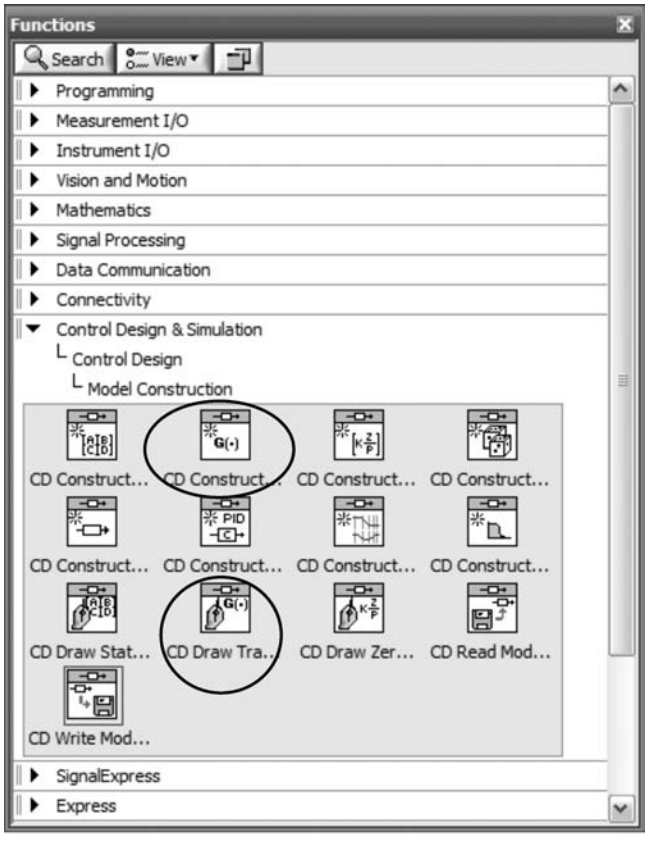
```
numg=100;
deng=[1 2 100];
'G(s)'
G=tf(numg,deng)
step(G);
title('Angular Velocity')
```

Analysis and design usually begins by selecting icons from the **Control Design** subpalette and dragging them to the **Block Diagram** window. The icons represent blocks of code and the cascading of code blocks can be thought of as a sequence of lines of code. Thus, an advantage of LabVIEW over MATLAB is that the programmer does not need to memorize coding language. For example, consider the MATLAB code shown in TryIt D.1 that produces the step response of $G(s) = 100/(s^2 + 2s + 100)$:

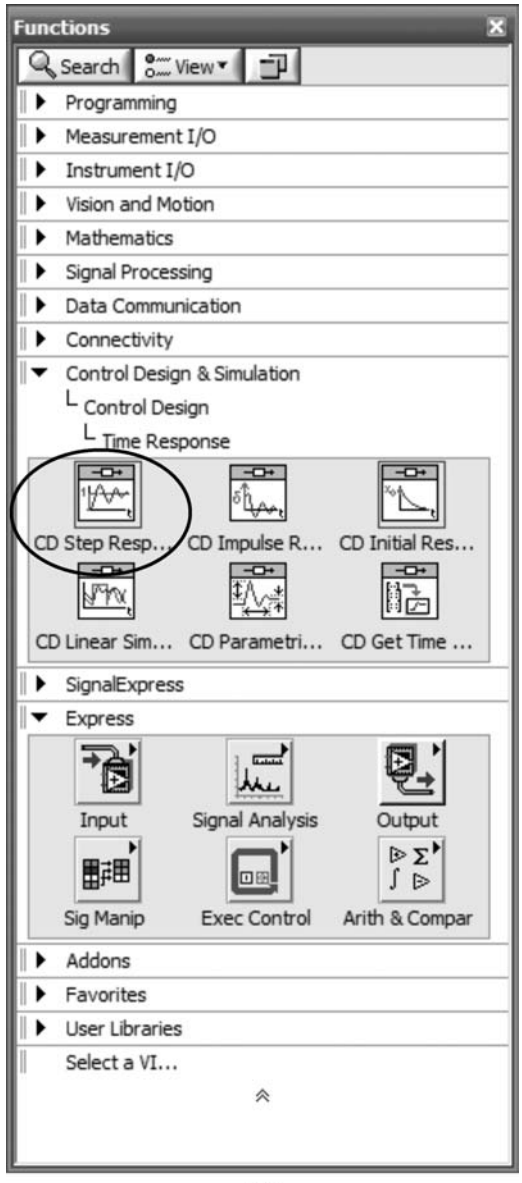
This step response can be produced in LabVIEW without knowing any coding language. We now demonstrate by following each step of Section D.3:

1. **Start LabVIEW** Start LabVIEW and select **Blank VI** from the window shown in Figure D.2.
2. **Select blocks** From the **Functions** palette, select the blocks shown in Figure D.3(a) and (b).
3. **Move/blocks to the Block Diagram window** Drag your icons one at a time to the **Block Diagram** window, Figure D.4.
4. **Obtain information about the block** Right-click each of the blocks and be sure the first two items under **Visible Items** are checked. Look at the **CD Construct Transfer Function Model.vi.** A **Polymorphic VI Selector** is shown at the bottom. Click the selector to bring up the menu. Select **SISO**. This block effectively creates the transfer function shown in the first four steps of the MATLAB code in TryIt D.1.

   Repeat for the **CD Draw Transfer Function Equation.vi** and select **TF** from the **Polymorphic VI Selector**. This block will write the transfer function symbolically in the display. Your selection from the polymorphic vi selector should match the format of the transfer function created by the **CD Construct Transfer Function Model.vi**.



(*a*)



(*b*)

FIGURE D.3 Selecting **a. CD Construct** . . . and **CD Draw** . . . ; **b. CD Step Response** . . .

Repeat for the **CD Step Response.vi.** and select **TF** from the **Polymorphic VI Selector**. This block will collect the data for the step response and permit plotting the data. This block effectively creates the last two commands of the MATLAB code shown in TryIt D.1.

5. **Interconnect and label blocks** You should now have the **Block Diagram** window shown in Figure D.4. Interconnect the code blocks. Click on the question mark on the right side of the toolbar to bring up the context menu. As your mouse passes above an icon, its context menu appears, showing the terminals. See Figure D.5. Interconnect the terminals by letting the mouse rest on a terminal until it becomes a spool of wire.

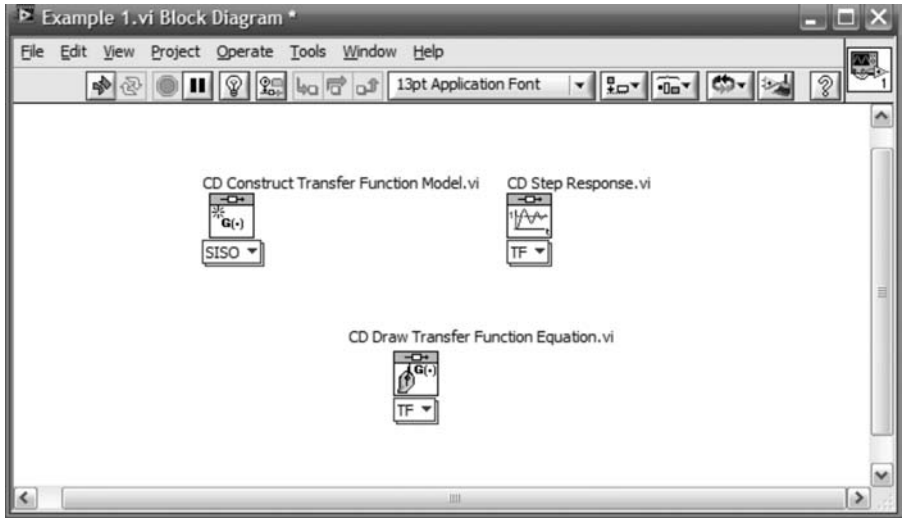


FIGURE D.4 **Block Diagram** window

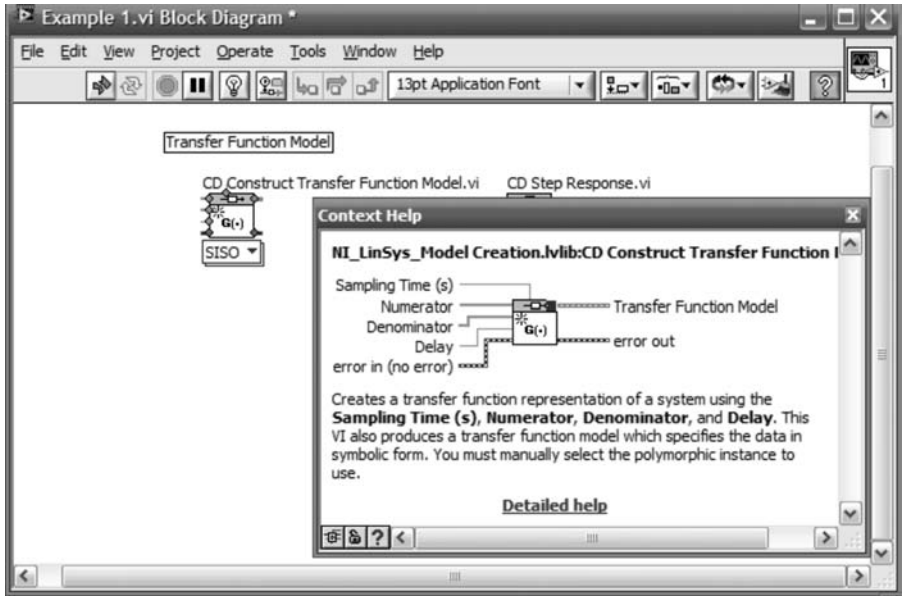


FIGURE D.5 **Context Help** for **CD Construct Transfer Function Model.vi**

Click on the terminal and then click on the destination terminal. The two terminals will appear as wired together. Continue wiring terminals until you have the **Block Diagram** window shown in Figure D.6. Mid-wire connections as shown can be made by letting your mouse rest at the connection point until it becomes a spool of wire.

6. **Create the interface to your block** You will now want to create the interface to specify parameters and view responses. This step will create the interface that will be accessed on the **Front Panel** window. The interfaces we will create are:

   - **CD Construct Transfer Function Model.vi** input parameter controls. Right-click on the numerator terminal shown in Figure D.5 and select **Create/Control**. Repeat for the denominator.

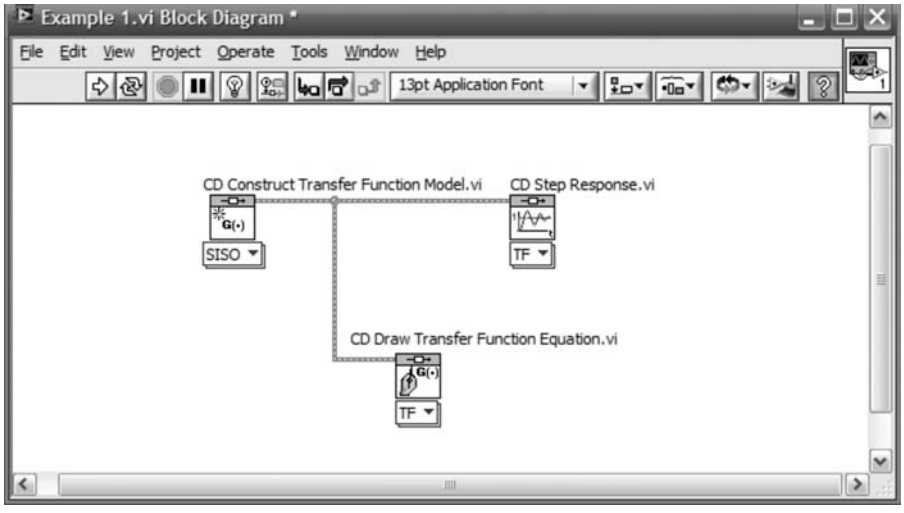
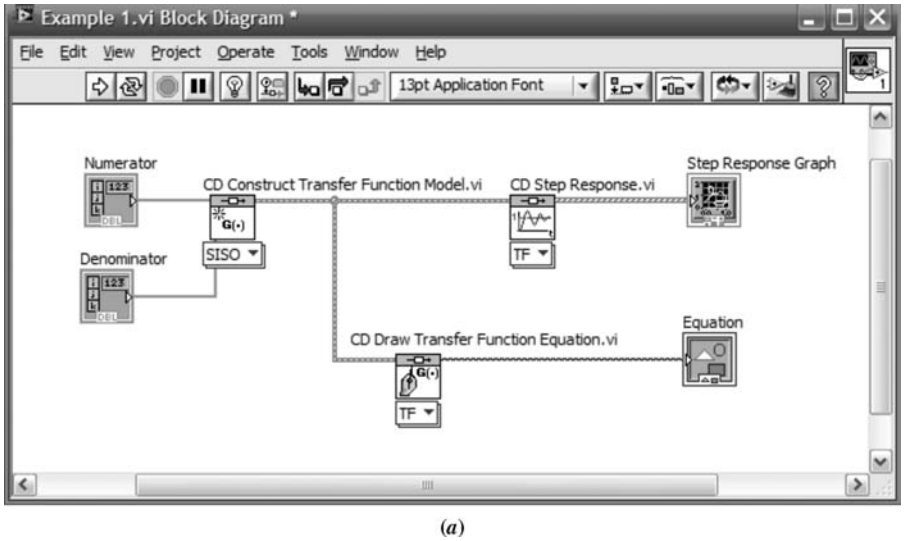


FIGURE D.6 Interconnected blocks

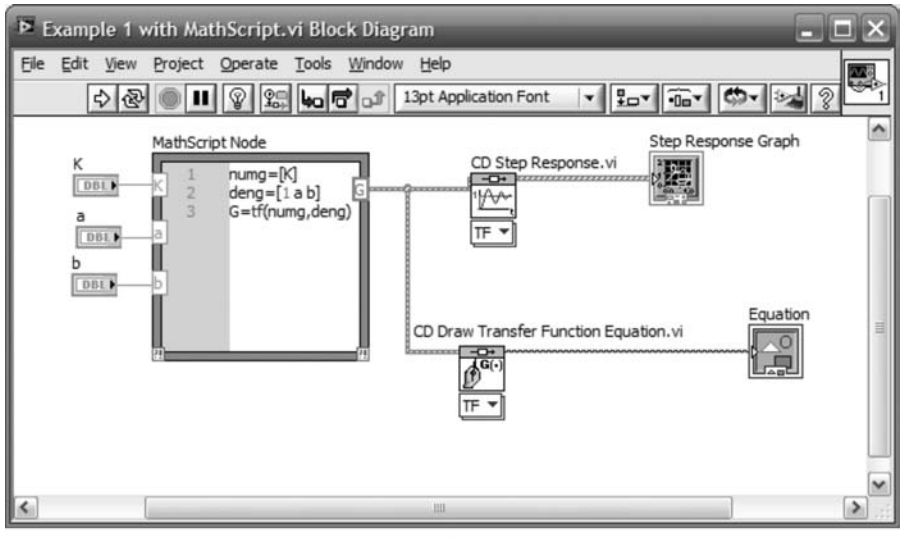
- **CD Step Response.vi** response plot indicator. Right-click on the **Step Response Graph** terminal and select **Create/Indicator**.
- **CD Draw Transfer Function Equation.vi** symbolic transfer function indicator. Right-click on the **Equation** terminal and select **Create/Indicator**. Your **Block Diagram** should now look similar to Figure D.7(*a*).

As an option, you can create transfer functions using a **MathScript** block if the MathScript RT Module is installed. This option is generally compatible with MATLAB's M-file code statements for creating your transfer function. Interfaces are then created to pass parameters to and from the M-file code. You should be familiar with MATLAB to use this option. The **MathScript** block is found in the **Programming/Structures/MathScript** palette. You create M-file code inside the **MathScript** block. Input and output interfaces are created and named identically



(*a*)

FIGURE D.7 **Block Diagram** window: **a**. with **Control Design** blocks and interfaces; (*figure continues*)



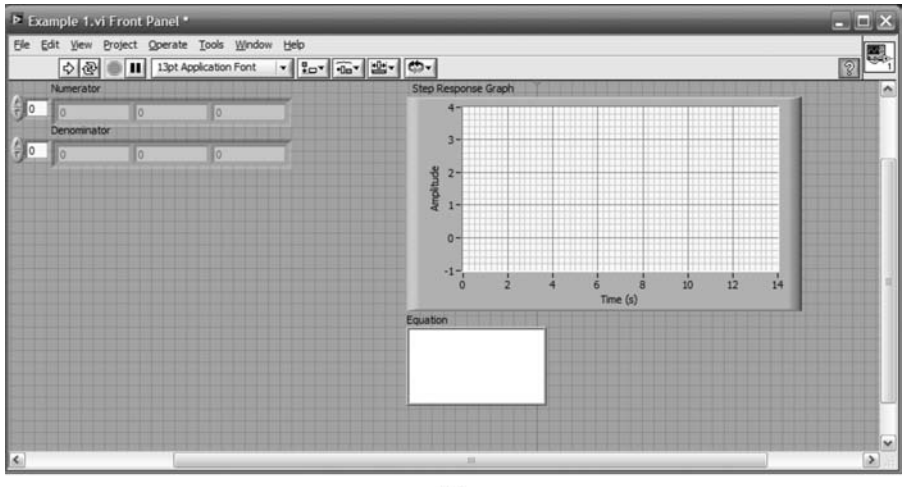
(*b*)

FIGURE D.7 (*Continued*) **b.** with **MathScript** block

to those within the M-file code. However, when using MathScript, you must create controls first on the **Front Panel**, rather than the **Block Diagram**. For example, to create the numeric interfaces for *K*, *a*, and b, right-click the **Front Panel** to produce the **Controls** palette. From this palette, produce each numeric control from **Modern/Numeric/Numeric Control**. The resulting controls are shown in Figure D.8(*b*). These interfaces are then wired to the appropriate terminals on the **Block Diagram**. Your **Block Diagram** should now look similar to Figure D.7(*b*).

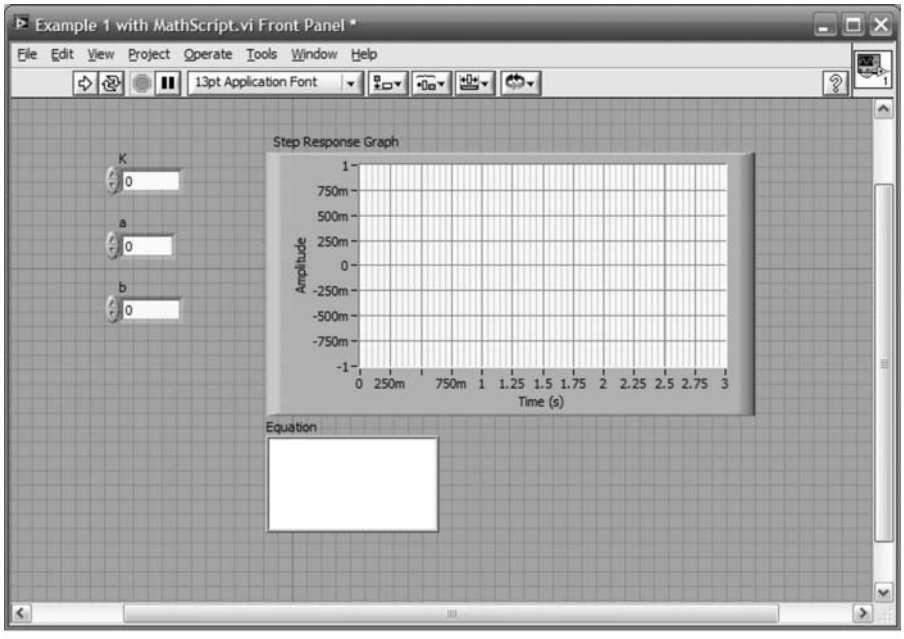
On the **Block Diagram** window menu bar, select **Window/Show Front Panel**. You will see the **Front Panel** shown in Figure D.8 created by your interfaces. You can double-click the labels above your interfaces either in the **Front Panel** window or the **Block Diagram** window to change the label to be more descriptive of your project.

7. **Set the controls** Using the **Front Panel** window, enter polynomial coefficients for the numerator and denominator in ascending order—lowest to highest. The selector to the left of the numerator and denominator shows the power of *s* for



(*a*)

FIGURE D.8 **Front Panel: a.** for **Block Diagram** shown in Figure D.7(*a*); (*figure continues*)



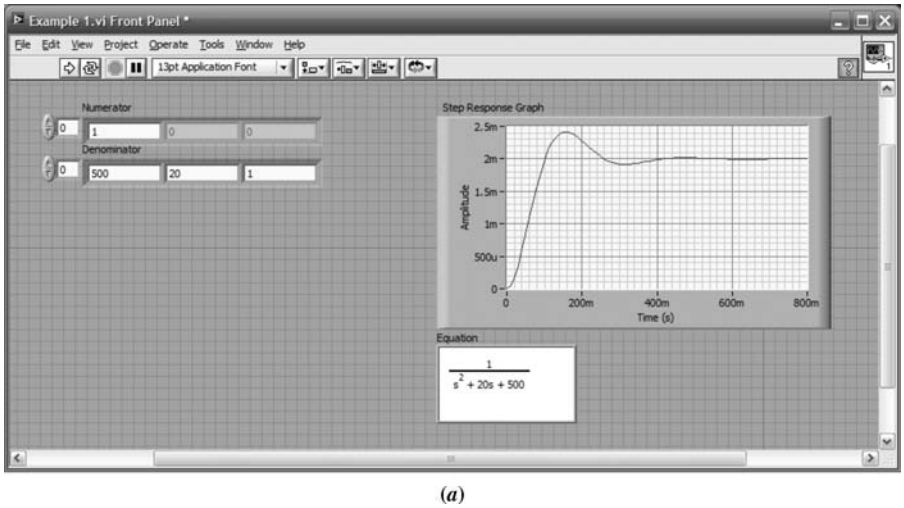
(*b*)

FIGURE D.8 (*Continued*) **b.** for **Block Diagram** shown in Figure D.7(*b*)

the left-most coefficient. Increasing the counter allows entry of higher-order coefficients not visible originally. To make all coefficients of a polynomial visible, let the mouse move on the right-hand edge of the polynomial indicator until the pointer becomes a double arrow and blue dots appear at the left and right edges of the entire polynomial indicator. You can then drag the right blue dot to expose more cells.

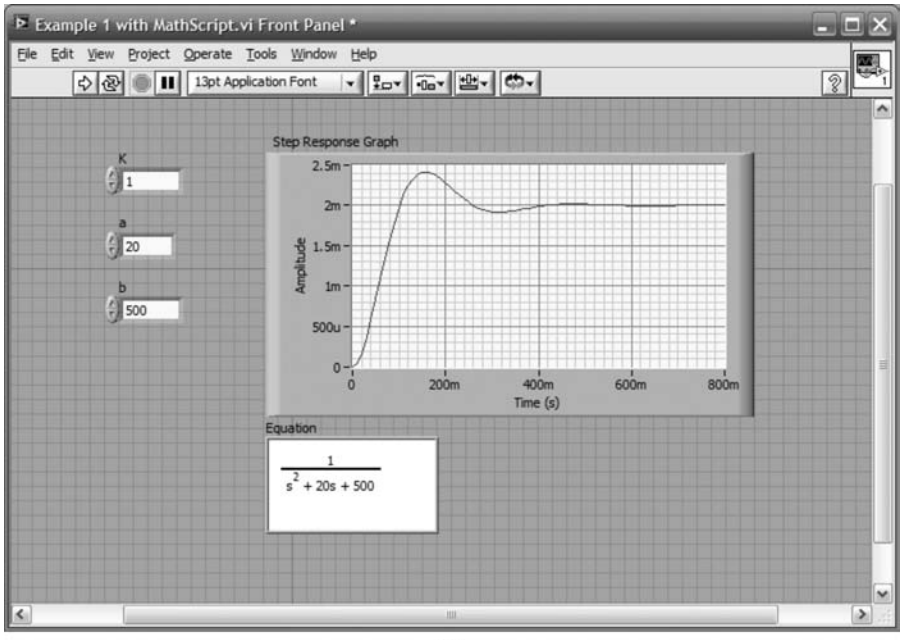
Familiarize yourself with the choices on the menu bar as well as those on the pop-up menus created when you right-click on any indicator or control. For example, under the **Edit** menu, among other choices, you can **Make Current Values Default** or **Reinitialize Values to Default**. Right-clicking the indicators or controls brings up a menu from which, among other choices, **Properties** can be selected to configure the indicator or control as desired.

8. **Run the program** Figure D.9 shows Example D.1 after execution. The figure shows the values entered, the equation, and the step response. Execution was initiated by clicking the arrow at the left of the toolbar.



(*a*)

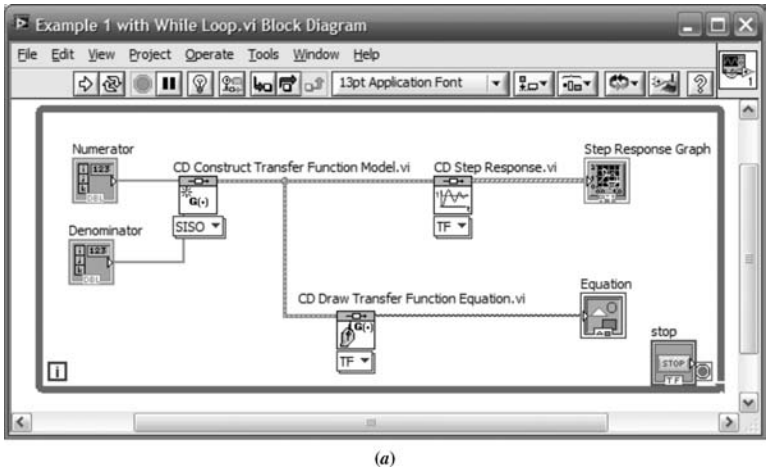
FIGURE D.9 **Front Panel** after execution: **a.** for block diagram in Figure 7(*a*); (*figure continues*)



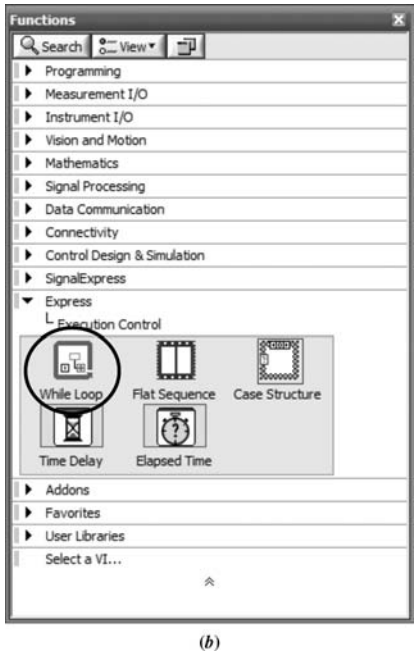
(*b*)

FIGURE D.9 (*Continued*) **b.** for block diagram in Figure 7(*b*)

The program can run continuously by clicking the curved arrows on the toolbar. Now, change values; hit the **Enter** key and see the results immediately displayed. Stop the program execution by clicking on the red hexagon on the toolbar. Another way of continuously running the program is to place a **While Loop** around the block diagram as shown in Figure D.10(*a*). The loop is accessed from **Functions/Express/Execution Control** as shown in Figure D.10(*b*). After



(*a*)



(*b*)

FIGURE D.10 **a.** Block diagram with **While Loop**; **b. Functions** palette showing **While Loop** location

selecting the **While Loop**, drag the cursor across the block diagram to create the continuous loop. A **stop** button will appear on the block diagram as well as on the **Front Panel**. At the lower right is a **Loop Interation** icon, which can be used to control the **While Loop**. The reader should consult the on-line documentation for further information.

Example D.2

Closed-Loop Step Response

In this example, we show how to display the step response of a unity-feedback system. For variety, we represent the open-loop system as a ratio of zeros over poles

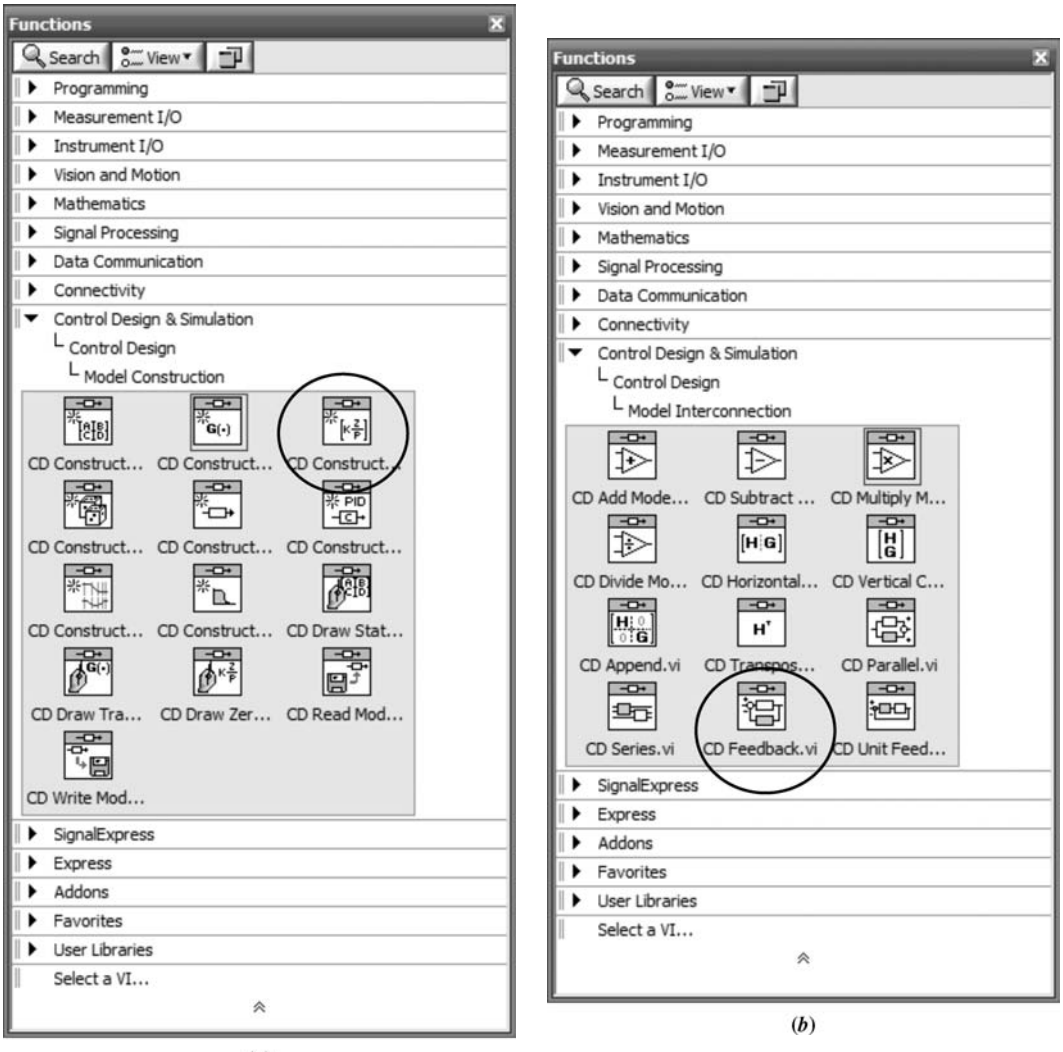
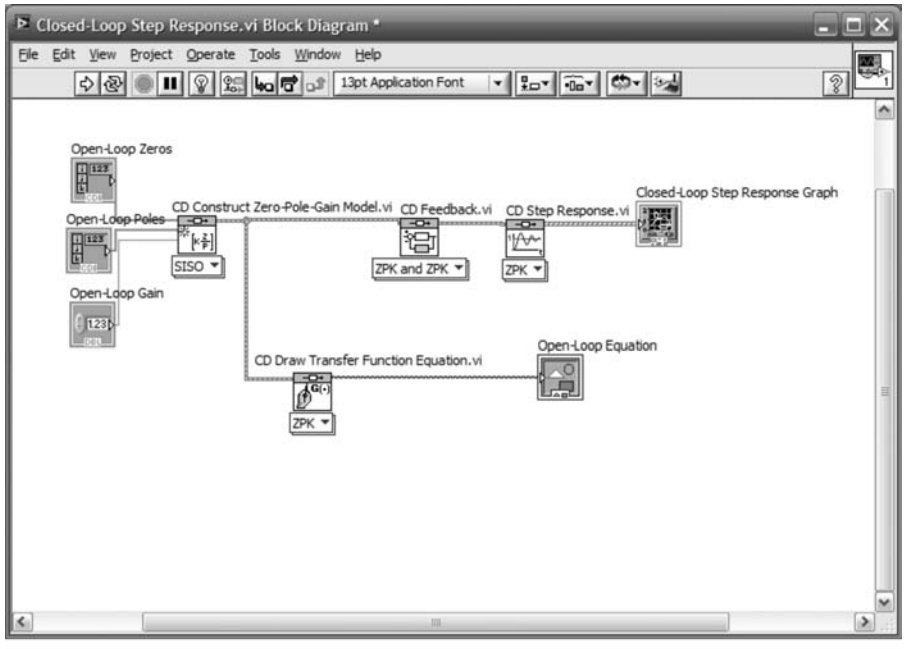


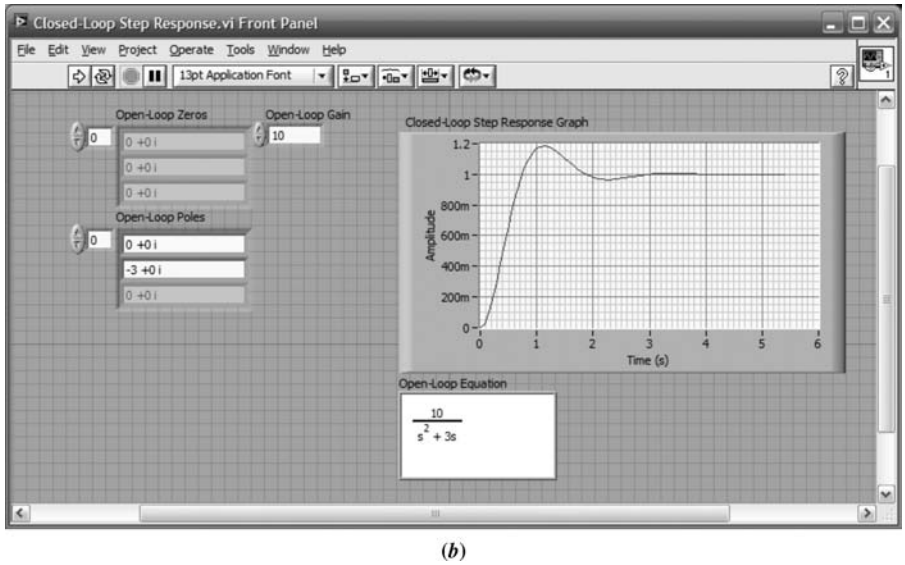
FIGURE D.11 **a.** Obtaining zero-pole-gain transfer function from the **Functions** palette; **b.** Obtaining **Feedback** interconnection from **Functions** palette

with a multiplying gain, analogous to MATLAB's **zpk** function. In the previous example, we represented the system as a ratio of polynomials, analogous to MATLAB's **tf** function.

1. **Select blocks** The zero-pole-gain transfer function is obtained from the **Functions** palette as shown in Figure D.11(*a*). We place this transfer function in the forward path of a unity-feedback system by following its block with a **Feedback** block obtained from the **Functions** palette as shown in Figure D.11(*b*).



(*a*)



(*b*)

FIGURE D.12 **a. Block Diagram** for Example D.2; **b. Front Panel** for Example D.2

If the **Model 2** input to the **Feedback** block is left unconnected, then a unity-feedback interconnection is assumed. Other options for interconnection, such as parallel and series, are shown on the palette of Figure D.11(*b*).

2. **Interconnect and label blocks** Producing the closed-loop step response is similar to Example D.1, except the step-response blocks are placed at the output of the **Feedback** block. The equation writer is wired to the system output as in Example D.1. All data types must be compatible and are shown selected with the pull-down menu at the base of the blocks. If you select **Automatic** in the pull-down menu, LabVIEW will select the correct form for you as you connect the blocks.

   The final **Block Diagram** and **Front Panel** for this example are shown in Figure D.12 (*a*) and (*b*), respectively. Notice that you enter open-loop poles, zeros, and gain on the **Front Panel** in place of polynomial open-loop numerator and denominator coefficients.

Example D.3

Root Locus Analysis and Design

We can obtain root locus plots by adding the **Root Locus** block obtained from the **Functions** palette as shown in Figure D.13. The **Root locus** block is connected to the output of the open-loop system and a **Root Locus Graph** indicator is formed at the output of the **Root Locus** block. The resultant **Block Diagram** and **Front Panel** are shown in Figure D.14(*a*) and (*b*) respectively.

Figure D.13 shows other characteristic blocks that can be added. For example, closed-loop poles and zeros, as well as damping ratio and natural frequency, can be displayed.

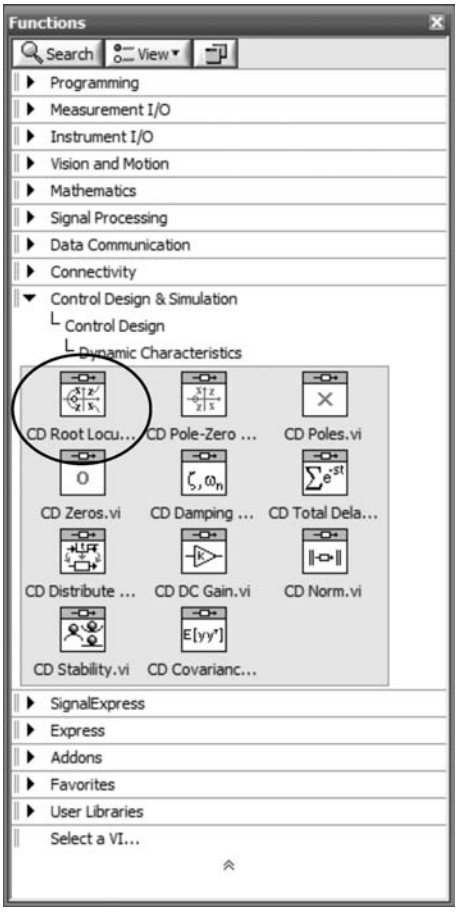
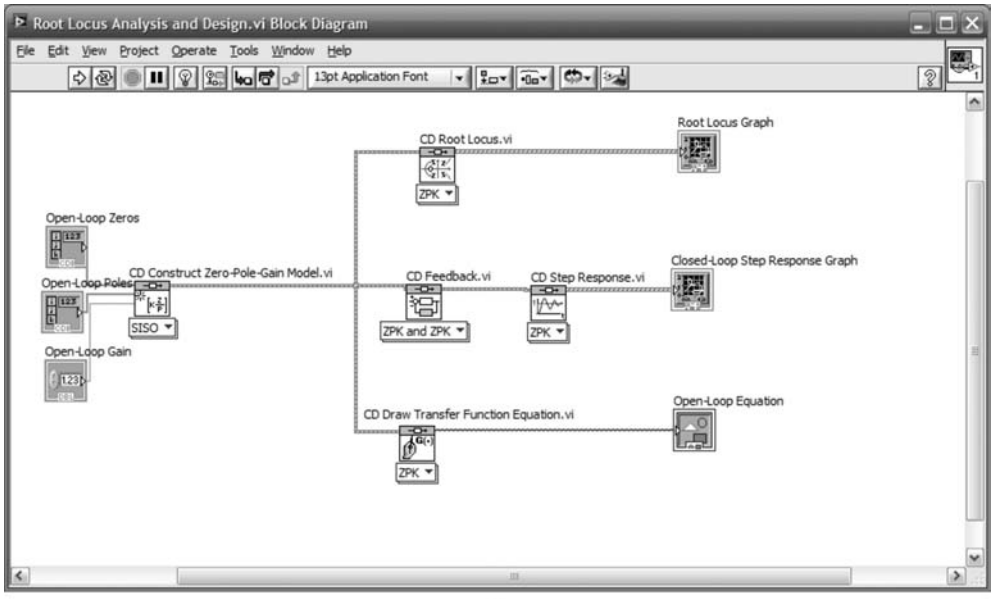
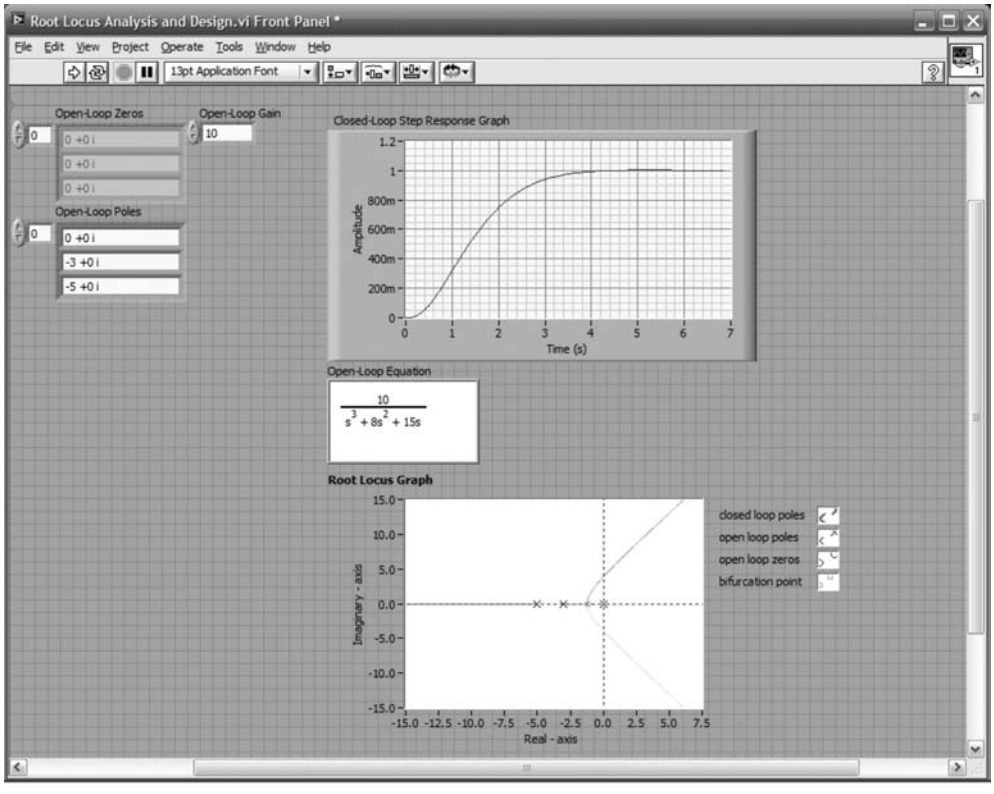


FIGURE D.13 **Functions** palette showing location of **Root Locus** block



(*a*)



(*b*)

FIGURE D.14 Windows showing root locus analysis: **a. Block Diagram**; **b. Front Panel**

Example D.4

Open- and Closed-Loop Sinusoidal Frequency Analysis and Design

We can obtain open- and closed-loop sinusoidal frequency response curves by replacing the **Root Locus** block with the **Bode** block to yield the open-loop frequency response. A copy of the **Bode** block can be added at the output of the **Feedback** block to obtain the closed-loop frequency response. Figure D.15 shows where to obtain the **Bode** block.

Figure D.16 shows the **Block Diagram** and **Front Panel** with open- and closed-loop Bode analysis. In order to display the plots, the indicators shown at the outputs of the **Bode** blocks were created.

Figure D.15 shows other alternatives for frequency response analysis. For example, in addition to the Bode plots, you can create an indicator telling you the gain and phase margins by using the **Gain and Phase Margin** block. Figure D.17 shows that result.

Finally, if you need to use Nyquist or Nichols charts, the associated blocks are shown in Figure D.15 and can replace the **Bode** blocks.

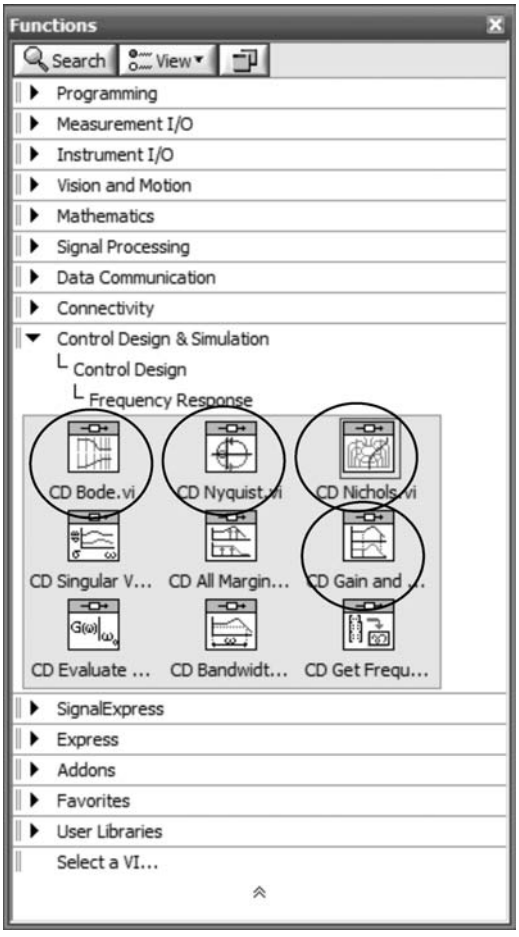
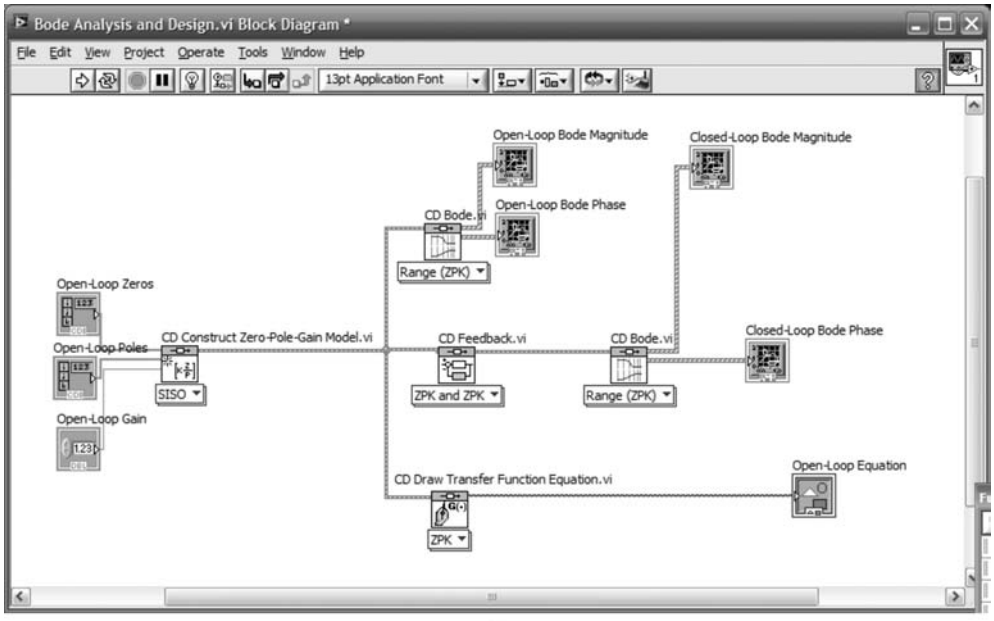
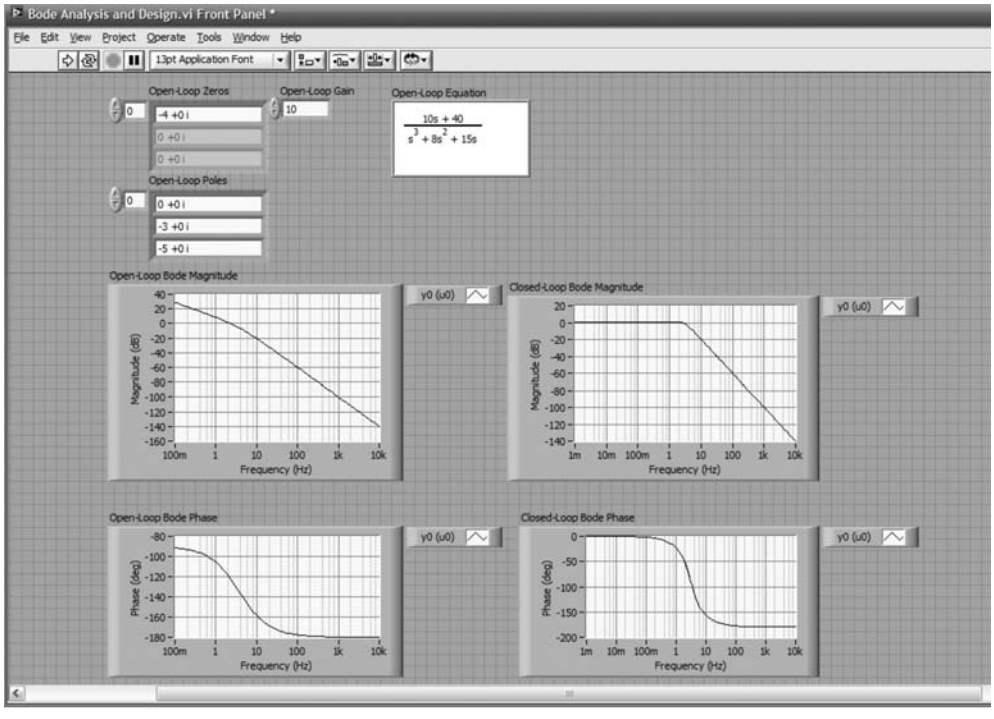


FIGURE D.15 **Functions** window showing frequency response blocks, such as **Bode**, **Nyquist**, **Nichols**, and **Gain and Phase Margin** blocks

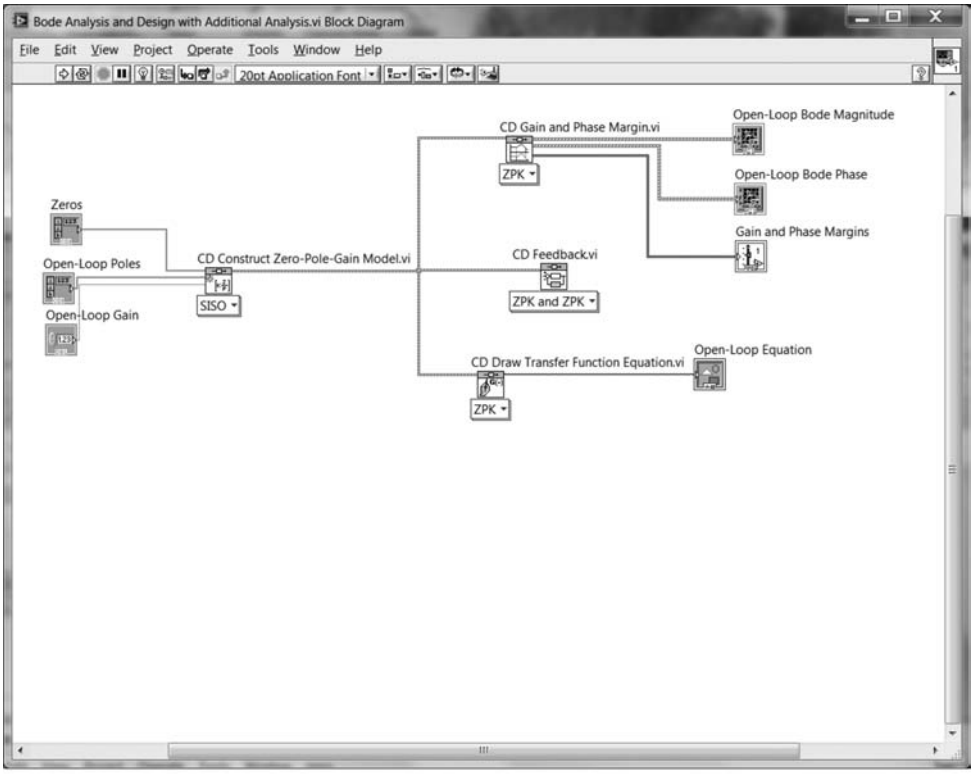


(*a*)

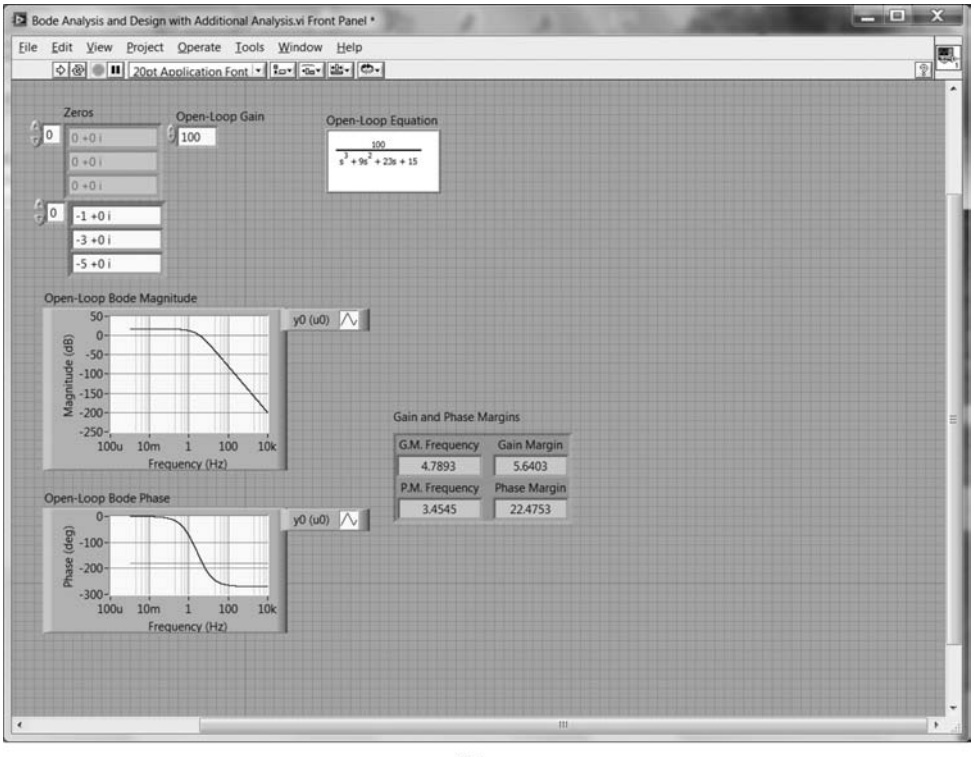


(*b*)

FIGURE D.16 Bode analysis via LabVIEW: **a. Block Diagram**; **b. From Panel**



(*a*)



(*b*)

FIGURE D.17 Bode analysis with gain and phase margin: **a. Block Diagram**; **b. Front Panel**

D.5 Simulation Examples

Whereas the LabVIEW block sequence for design and analysis is analogous to following the code statement sequence in a MATLAB M-file, the LabVIEW block sequence for simulation is analogous to following the block sequence of a Simulink diagram.

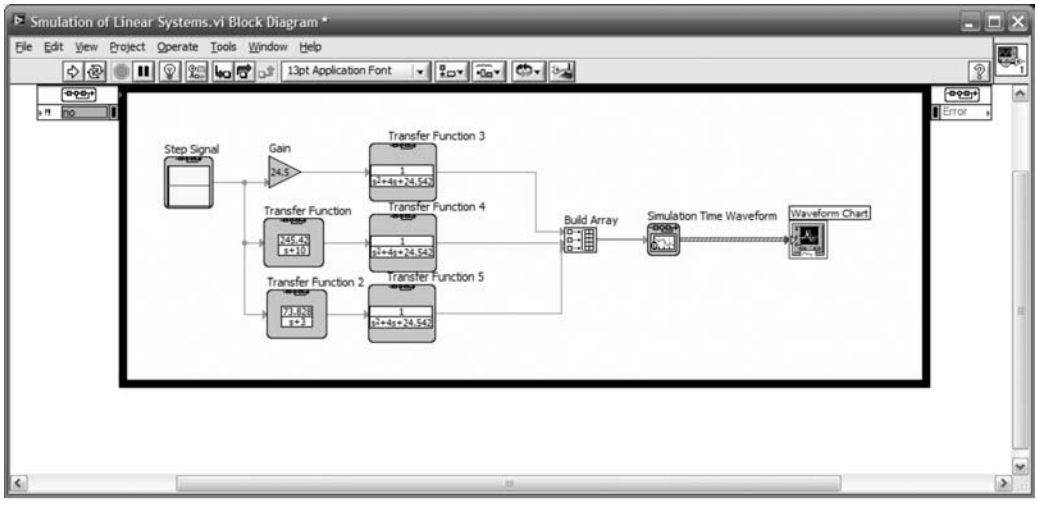
In this section, we show examples of simulation using LabVIEW. For control system simulation, icons for the block diagram are taken from the **Simulation** subpalette under the **Control Design and Simulation** palette. Our examples will parallel the examples shown in Appendix C which uses Simulink.

Example D.5

Simulation of Linear Systems

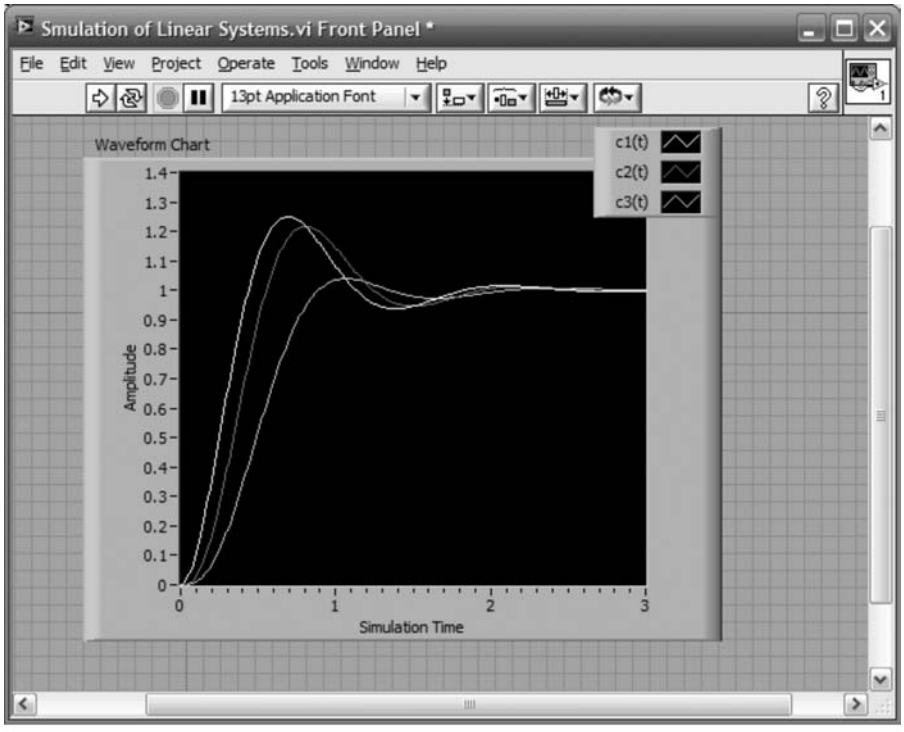
Create Block Diagram and Front Panel Figure D.18 shows the **Block Diagram** and **Front Panel** for simulating a linear system. The simulation reproduces Example C.1 in Appendix C, which uses Simulink. Blocks are selected from the **Simulation** subpalette under the **Control Design and Simulation** palette and must be placed within the **Simulation Loop** obtained from **Functions/Control Design and Simulation/Simulation/Simulation Loop**. We now enumerate the detailed steps required to create the **Block Diagram** and **Front Panel:**

1. Transfer functions are obtained from **Functions/Control Design and Simulation/Simulation/Continuous Linear Systems/Transfer Function**. Right-click on each transfer function and select **Configuration** to enter the parameter values shown in Figure D.18(*a*) or equivalently in Figure C.5
2. The gain block is obtained from **Functions/Control Design and Simulation/Simulation/Signal Arithmetic/Gain**. Right-click on the gain block and select **Configuration** to enter the parameter value.



(*a*)

FIGURE D.18 Simulation of linear systems: **a. Block Diagram**; (*figure continues*)



(*b*)

FIGURE D.18 (*Continued*) **b. Front Panel**

3. The step-input block is obtained from **Functions/Control Design and Simulation/Simulation/Signal Generation/Step Signal**. Right-click on the gain block and select **Configuration** to enter the parameter value.
4. In order to display the three step-response curves simultaneously, we use a **Build Array** block obtained from **Functions/Programming/Array/Build Array**. Drag the bottom of the icon to expose the correct number of inputs three for this case).
5. To create the display, we use the **Simulation Time Waveform** block obtained from **Functions/Control Design and Simulation/Simulation/Graph Utilities/ Simtime Waveform**. Right-click the output of the **Simtime Waveform** block and select **Create/Indicator** to produce the **Waveform Chart** icon and the **Front Panel** display.

Configure simulation loop Finally, set the simulation parameters by right-clicking the **Simulation Loop** and selecting **Configure Simulation Parameters** . . . Set the parameters as shown in Figure D.19.

Configure graph parameters On the **Front Panel**, right-click the graph and select **Properties** to configure graph parameters if required. Select the legend and expand it vertically to expose all three plot identities. The titles in the legend can be changed to reflect meaningful labels for the plots.

Run the simulation Perform the simulation by clicking the arrow at the extreme left of the toolbar on the **Front Panel** window. You can erase curves between trials by right-clicking the display and selecting **Data Operations/Clear Chart**.

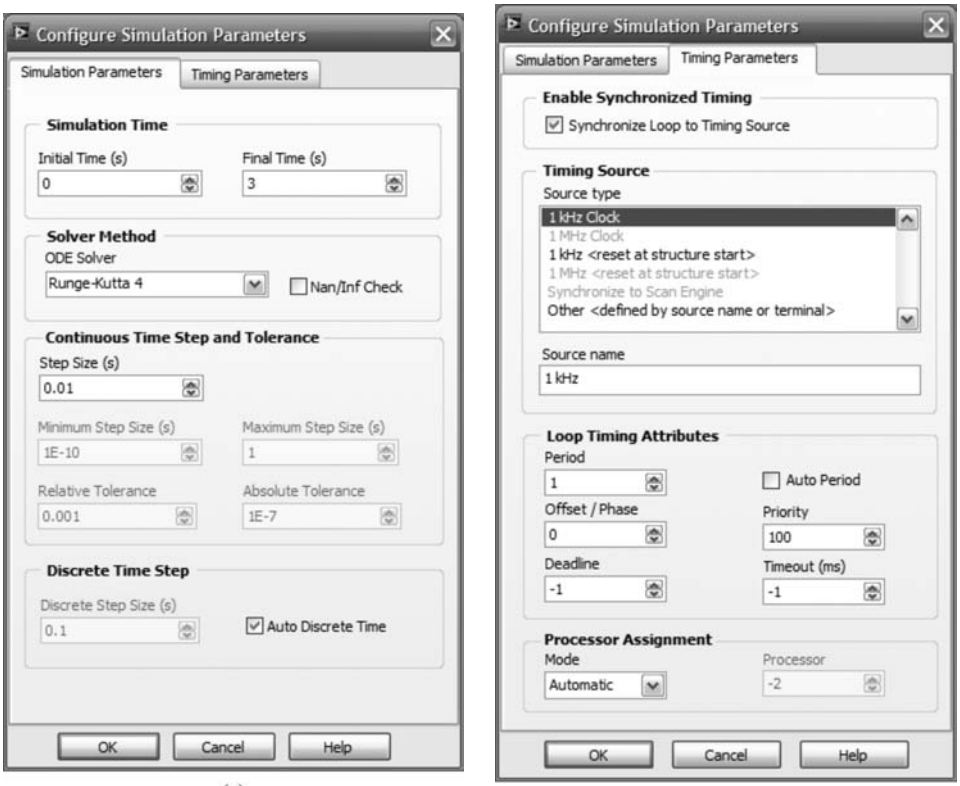
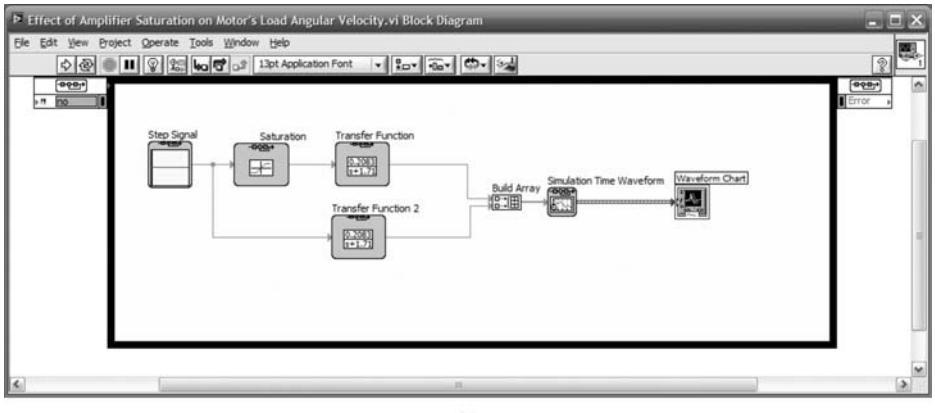


FIGURE D.19 Configuring the **Simulation Loop** parameters: **a.** Simulation parameters; **b.** Timing parameters

Example D.6

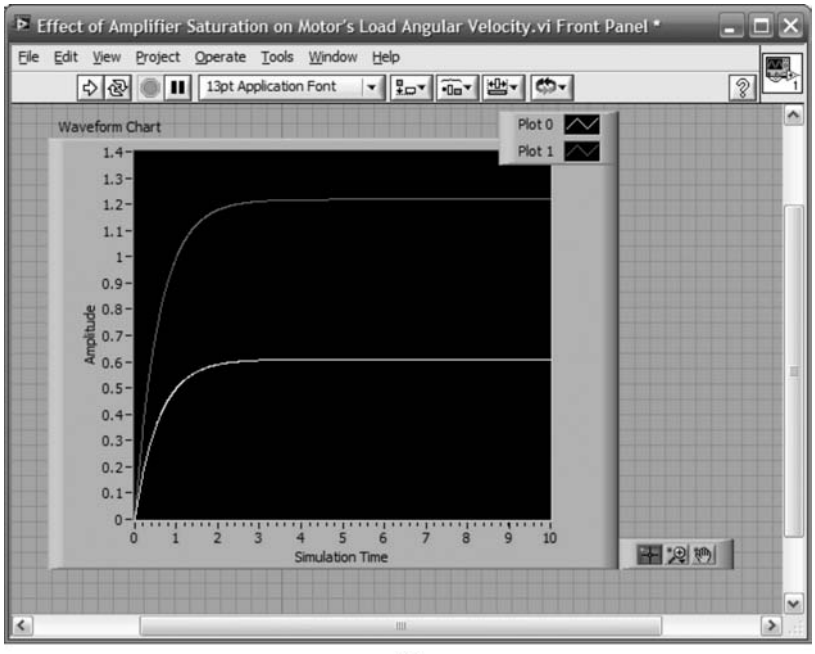
Effect of Amplifier Saturation on Motor's Load Angular Velocity

Create Block Diagram and Front Panel The **Block Diagram and Front Panel** for simulating a dc motor with and without saturation are shown in Figure D.20. The **Saturation** block is obtained from **Control Design & Simulation/Simulation/Nonliner Systems/Saturation**.



(a)

FIGURE D.20 Simulation of a dc motor with and without saturation: **a. Block Diagram**; (*figure continues*)



(*b*)

FIGURE D.20 (*Continued*) **b. Front Panel**

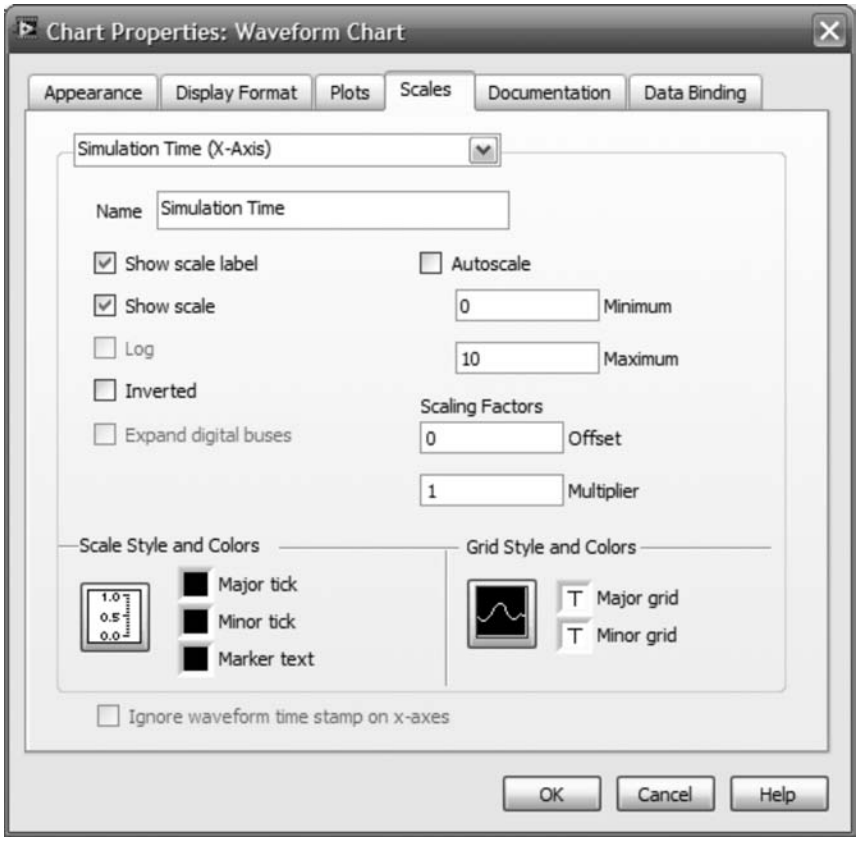


FIGURE D.21 Chart Properties: Waveform Chart Window

Configure simulation loop Configure the simulation loop as shown in figure D.19, except change the **Final Time (s)** in Figure D.19(*a*) to 10.

Configure graph parameters On the **Front Panel**, right click the graph and select **Properties** to configure graph parameters. Select the **Scales** tab and enter 10 in the **Maximum** box as shown in Figure D.21. Select the legend and expand it vertically to expose both plot identities. The titles in the legend can be changed to reflect meaningful labels for the plots.

Run the simulation Perform the simulation by clicking the arrow at the extreme left of the toolbar on the **Front Panel** window. You can erase curves between trials by right-clicking the display and selecting **Data Operations/Clear Chart**.

Example D.7

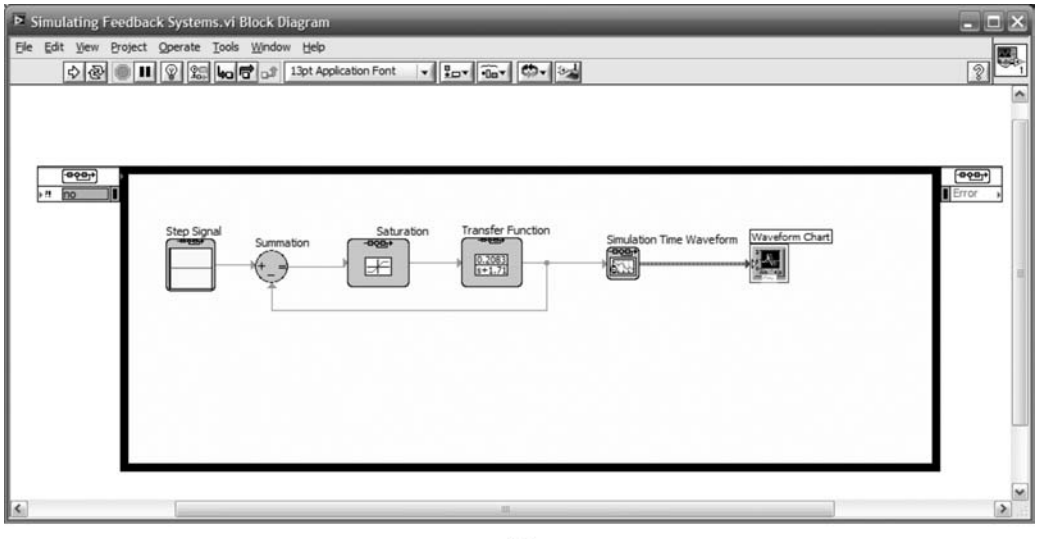
Simulating Feedback Systems

Create Block Diagram and Front Panel The **Block Diagram** and **Front Panel** for simulating feedback systems is shown in Figure D.22. The **Summation** block is obtained from **Control Design & Simulation/Simulation/Signal Arithmetic/Summation**.

Configure Summation and other blocks Right-click the **Summation** block and select **Configuration** . . . Repeat for other blocks.

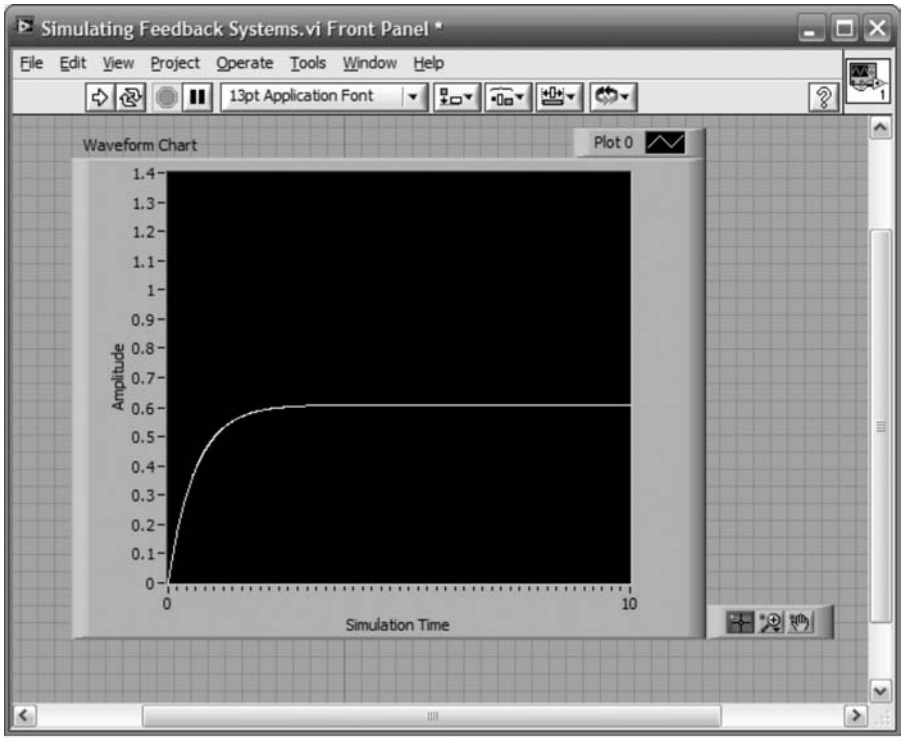
Configure simulation loop Configure the simulation loop as shown in Figure D.19, except change the **Final Time (s)** in Figure D.19(*a*) to 10.

Configure graph parameters On the **Front Panel**, right click the graph and select **Properties** to configure graph parameters. Select the **Scales** tab and enter 10 in the **Maximum** box as shown in Figure D.21.



(*a*)

FIGURE D.22 Simulation of feedback systems: **a. Block Diagram**; (*figure continues*)



(*b*)

FIGURE D.22 (*Continued*) **b. Front Panel**

Run the simulation Perform the simulation by clicking the arrow at the extreme left of the toolbar on the **Front Panel** window. You can erase curves between trials by right-clicking the display and selecting **Data Operations/Clear Chart**.

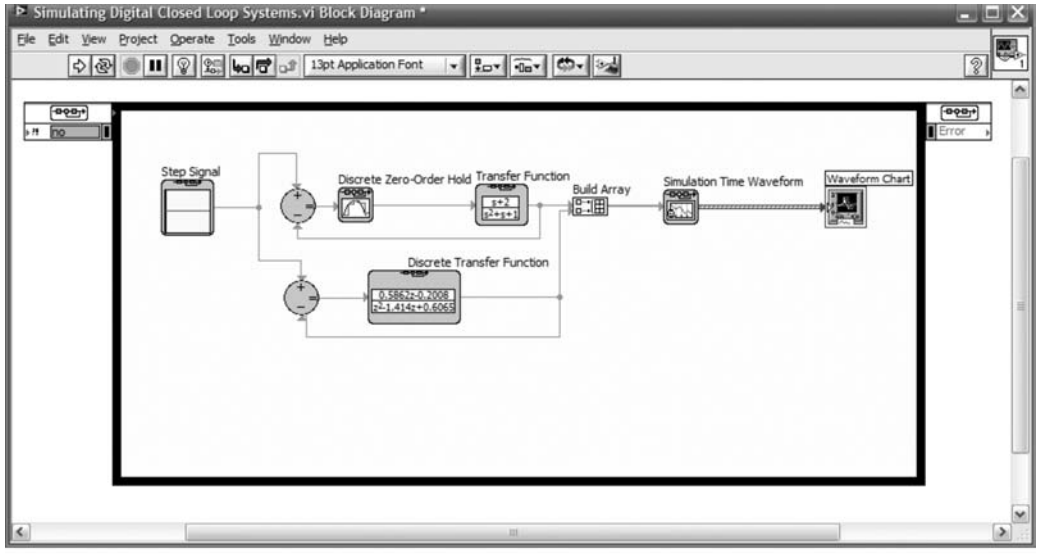
Example D.8

Simulating Digital Systems with the Simulation Palette

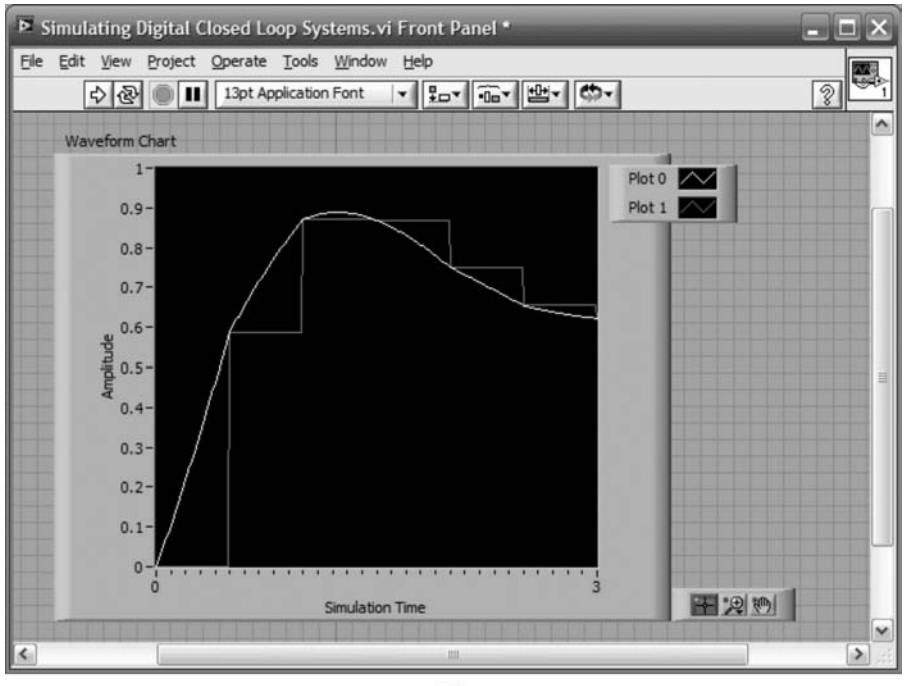
Digital systems, such as Example C.4 in Appendix C, can be simulated using LabVIEW. However, there are restrictions on the transfer functions used in the simulation. LabVIEW requires that all inputs to the transfer functions be present at the beginning of the simulation or else a cycle error will result. Unfortunately, this requirement limits the use of transfer functions to those with a denominator of higher order than the numerator. Under these conditions, the reader is advised to use either MATLAB or the **Control Design** palette rather than the **Simulation** palette of the **Control Design & Simulation** function.

Our first digital example will simulate a digital feedback system using the **Simulation** palette with proper transfer functions. The next example will simulate Example C.4 in Appendix C, which does not have proper transfer functions, using LabVIEW's **Control Design** palette.

Create Block Diagram and Front Panel The **Block Diagram** and **Front Panel** for simulating digital systems is shown in Figure D.23. The **Discrete Zero-Order Hold** block is obtained from **Control Design & Simulation/Simulation/Discrete Linear Systems/Discrete Zero-Order Hold**. The **Discrete Transfer Function** is obtained



(*a*)



(*b*)

FIGURE D.23 Simulation of digital systems with **Simulation** palette: **a. Block Diagram**; **b. Front Panel**

from **Control Design & Simulation/Simulation/Discrete Linear Systems/Discrete Transfer Function**.

Configure Discrete Zero-Order Hold and other blocks Right click the **Discrete Zero-Order Hold** block and select **Configuration . . .** Set the sample period to 0.5 second. Configure the transfer functions as shown on the **Block Diagram**. Configure the **Step Signal** to be a unit step.

Configure simulation loop Configure the simulation loop as shown in Figure D.19.

Configure graph parameters On the **Front Panel**, right click the graph and select **Properties** to configure graph parameters. Select the **Scales** tab and enter three in the **Maximum** box for both the *x*-and *y*-axes as shown in Figure D.21. Select the legend and expand it vertically to expose both plot identities. The titles in the legend can be changed to reflect meaningful labels for the plots.

Run the simulation Perform the simulation by clicking the arrow at the extreme left of the toolbar on the **Front Panel** window. You can erase curves between trials by right-clicking the display and selecting **Data Operations/Clear Chart**.

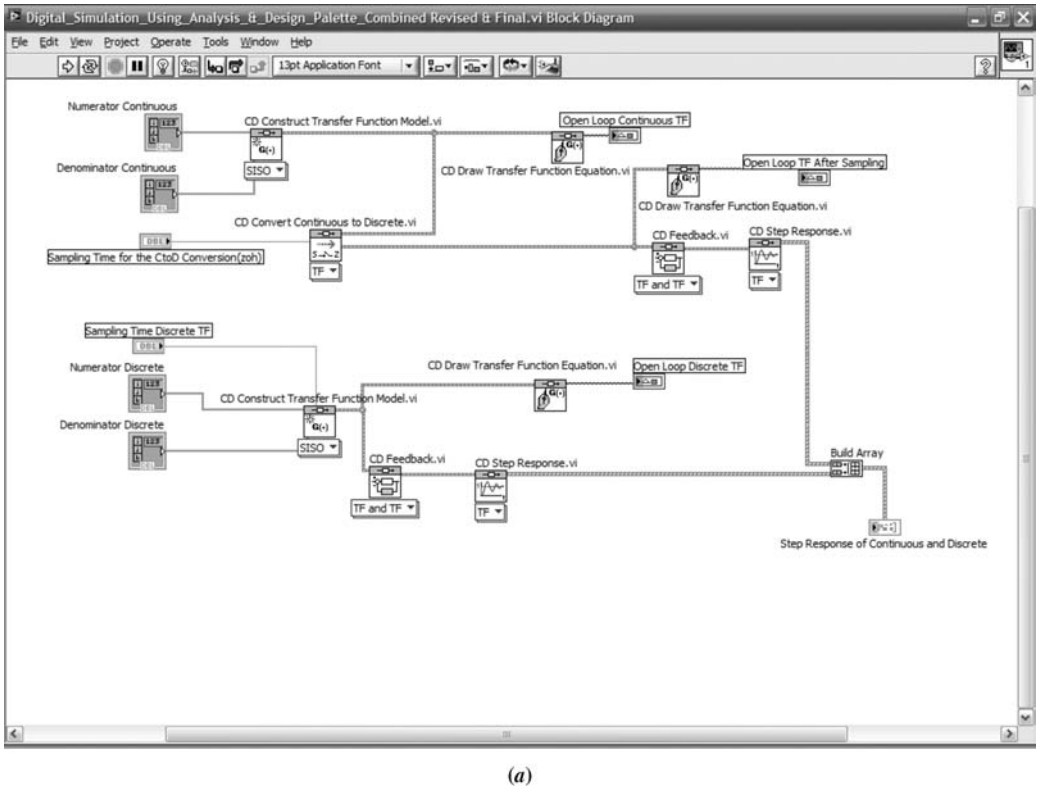
The simulation shows the difference in responses obtained by (1) modeling the digital system as a zero-order hold cascaded with a linear system (Plot 0), or (2) modeling the system with a digital transfer function (Plot 1).

Example D.9

Simulating Digital Systems with the Control Design Palette

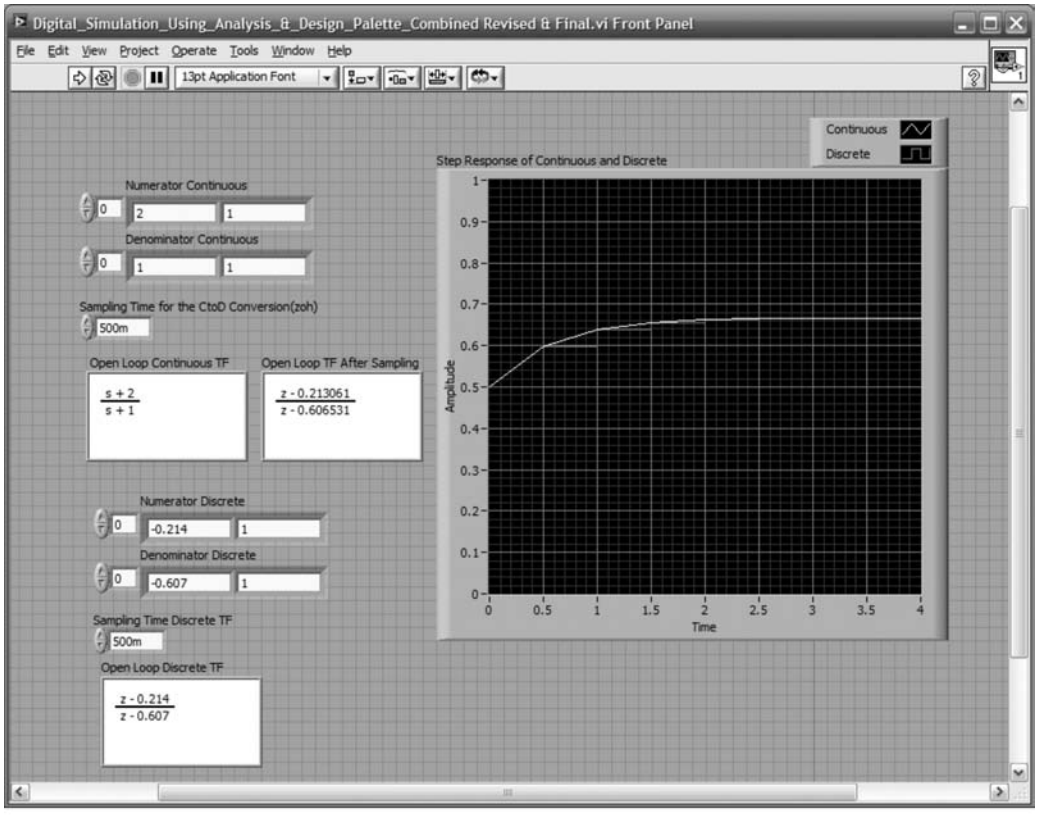
In order to avoid cycle errors in LabVIEW, we use the **Control Design** palette when we have transfer functions for which the numerator and denominator are of the same order. This example reproduces Simulink Example C.4.

Create Block Diagram and Front Panel The **Block Diagram** and **Front Panel** for this example are shown in Figure D.24. Wire the blocks as shown.



(*a*)

FIGURE D.24 Simulation of digital systems with **the Control Design** palette: **a. Block Diagram**; (*figure continues*)



(*b*)

FIGURE D.24 (*Continued*) **b. Front Panel**

Most of the blocks were previously discussed in Example D.1 and D.2. Digital transfer functions are created using the same blocks as continuous systems, but with a nonzero **Sampling Time(s)** input.

The **CD Convert Continuous to Discrete.vi**, is obtained from **Functions/Control Design & Simulation/Control Design/Model Conversion/CD Convert Continuous to Discrete.vi**.

The **Build Array** is obtained from **Functions/Programming/Array/Build Array**. Expand the **Build Array** block to show two inputs.

Configure parameters for Build Array Right-click on **Build Array** and select **Concatenate Inputs**. Right-click again on **Build Array** and select **Create/Indicator**.

Right-click the indicator on the front panel and select **Replace**. Using the resulting palettes as shown in Figure D.25, select the **XY Graph**.

On the front panel expand the legend to show two graphs. Title the legend components as shown. Change the *x*- and *y*-axes' starting and ending points as desired by right-clicking the graph and selecting **Properties**. In the **Properties** window, select **Scales** and enter the desired information.

Right-click the graph on the front panel and select **Data Operations** and make your current values the default. Also, right-click again and choose to reinitialize to your default values. You may also choose to clear the current plot.

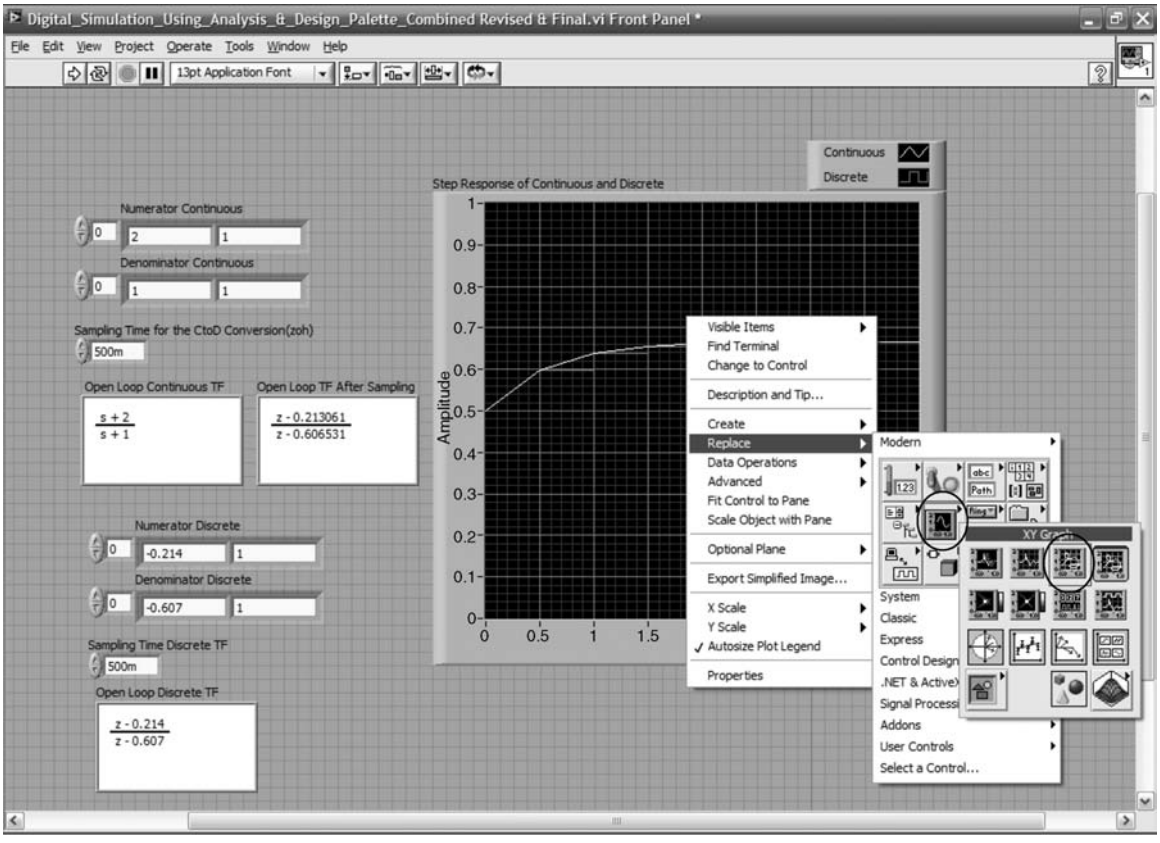


FIGURE D.25 Choosing **XYGraph**

Configure parameters for CD Convert Continuous to Discrete.vi Right-click and create a control for **Sample Time(s), Numerator**, and **Denominator** as described in Example D.1. Set the values as shown on the **Front Panel**.

Configure parameters for CD Construct Transfer Function Model.vi as a discrete model Right-click and create a control for **Sample Time(s)**, **Numerator**, and **Denominator** as described in Example 1. Set the values as shown on the **Front Panel**.

Configure parameters for all CD Draw Transfer Function Equation.vi Right-click and create a control for **Equation** as described in Example D.1. Set the values as shown on the **Front Panel**.

Run simulation See Example D.1 for a description. The results are shown in Figure D.24(*b*).

Summary

This appendix presented LabVIEW as an alternative to MATLAB for analysis, design, and simulation. Our discussion was divided into analysis and design, and simulation.

Analysis and design is performed by interconnecting code blocks, which is analogous to writing in-line code for MATLAB M-files. Since the LabVIEW code blocks are represented by icons, an advantage of using LabVIEW is that you do not have to know specific code statements.

Simulation is performed by interconnecting code blocks and is analogous to Simulink flow diagrams.

LabVIEW has more extensive applications than those covered here. You can create virtual instruments on your computer monitor that can operate external hardware and transmit and receive telemetric data. It is left to the interested reader to pursue these advanced topics.

Bibliography

National Instruments. *Getting Started with LabVIEW*. National Instruments, Austin, TX. 2003–2007.

National Instruments. *LabVIEW Fundamentals*. National Instruments, Austin, TX. 2003–2007.

National Instruments. *LabVIEW™ Control Design User Manual*. National Instruments, Austin, TX. 2004–2008.

National Instruments. *Introduction to LabVIEW in 3 Hours for Control Design and Simulation*. National Instruments Course Notes, Austin, TX.

Glossary

Acceleration constant $\lim_{s \to 0} s^2 G(s)$

Actuating signal The signal that drives the controller. If this signal is the difference between the input and output, it is called the *error*.

Analog-to-digital converter A device that converts analog signals to digital signals.

Armature The rotating member of a dc motor through which a current flows.

Back emf The voltage across the armature of a motor.

Bandwidth The frequency at which the magnitude frequency response is -3 dB below the magnitude at zero frequency.

Basis Linearly independent vectors that define a space.

Bilinear transformation A mapping of the complex plane where one point, s, is mapped into another point, z, through $z = (as + b)/(cs + d)$.

Block diagram A representation of the interconnection of subsystems that form a system. In a linear system, the block diagram consists of blocks representing subsystems, arrows representing signals, summing junctions, and pickoff points.

Bode diagram (plot) A sinusoidal frequency response plot where the magnitude response is plotted separately from the phase response. The magnitude plot is dB versus log ω, and the phase plot is phase versus log ω. In control systems, the Bode plot is usually made for the open-loop transfer function. Bode plots can also be drawn as straight-line approximations.

Branches Lines that represent subsystems in a signal-flow graph.

Break frequency A frequency where the Bode magnitude plot changes slope.

Breakaway point A point on the real axis of the s-plane where the root locus leaves the real axis and enters the complex plane.

Break-in point A point on the real axis of the s-plane where the root locus enters the real axis from the complex plane.

Characteristic equation The equation formed by setting the characteristic polynomial to zero.

Characteristic polynomial The denominator of a transfer function. Equivalently, the unforced differential equation, where the differential operators are replaced by s or λ.

Classical approach to control systems *See* **frequency domain techniques**.

Closed-loop system A system that monitors its output and corrects for disturbances. It is characterized by feedback paths from the output.

Closed-loop transfer function For a generic feedback system with $G(s)$ in the forward path and $H(s)$ in the feedback path, the closed-loop transfer function, $T(s)$, is $G(s)/[1 \pm G(s)H(s)]$, where the + is for negative feedback, and the – is for positive feedback.

Compensation The addition of a transfer function in the forward path or feedback path for the purpose of improving the transient or steady-state performance of a control system.

Compensator A subsystem inserted into the forward or feedback path for the purpose of improving the transient response or steady-state error.

Constant *M* circles The locus of constant, closed-loop magnitude frequency response for unity feedback systems. It allows the closed-loop magnitude frequency response to be determined from the open-loop magnitude frequency response.

Constant *N* circles The locus of constant, closed-loop phase frequency response for unity feedback systems. It allows the closed-loop phase frequency response to be determined from the open-loop phase frequency response.

Controllability A property of a system by which an input can be found that takes every state variable from a desired initial state to a desired final state in finite time.

Controlled variable The output of a plant or process that the system is controlling for the purpose of desired transient response, stability, and steady-state error characteristics.

Controller The subsystem that generates the input to the plant or process.

Critically damped response The step response of a second-order system with a given natural frequency that is characterized by no overshoot and a rise time that is faster than any possible overdamped response with the same natural frequency.

Damped frequency of oscillation The sinusoidal frequency of oscillation of an underdamped response.

Damping ratio The ratio of the exponential decay frequency to the natural frequency.

Decade Frequencies that are separated by a factor of 10.

Decibel (dB) The decibel is defined as $10 \log P_G$, where P_G is the power gain of a signal. Equivalently, the decibel is also $20 \log V_G$, where V_G is the voltage gain of a signal.

Decoupled system A state-space representation in which each state equation is a function of only one state variable. Hence, each differential equation can be solved independently of the other equations.

Digital compensator A sampled transfer function used to improve the response of computer-controlled feedback systems. The transfer function can be emulated by a digital computer in the loop.

Digital-to-analog converter A device that converts digital signals to analog signals.

Disturbance An unwanted signal that corrupts the input or output of a plant or process.

Dominant poles The poles that predominantly generate the transient response.

Eigenvalues Any value, λ_i, that satisfies $\mathbf{AX_i} = \lambda_i \mathbf{x_i}$ for $\mathbf{x_i} \neq 0$. Hence, any value, λ_i, that makes $\mathbf{x_i}$ an eigenvector under the transformation $\mathbf{A}$.

Eigenvector Any vector that is collinear with a new basis vector after a similarity transformation to a diagonal system.

Electric circuit analog An electrical network whose variables and parameters are analogous to another physical system. The electric circuit analog can be used to solve for variables of the other physical system.

Electrical admittance The inverse of electrical impedance. The ratio of the Laplace transform of the current to the Laplace transform of the voltage.

Electrical impedance The ratio of the Laplace transform of the voltage to the Laplace transform of the current.

Equilibrium The steady-state solution characterized by a constant position or oscillation.

Error The difference between the input and the output of a system.

Euler's approximation A method of integration where the area to be integrated is approximated as a sequence of rectangles.

Feedback A path through which a signal flows back to a previous signal in the forward path in order to be added or subtracted.

Feedback compensator A subsystem placed in a feedback path for the purpose of improving the performance of a closed-loop system.

Forced response For linear systems, that part of the total response function due to the input. It is typically of the same form as the input and its derivatives.

Forward-path gain The product of gains found by traversing a path that follows the direction of signal flow from the input node to the output node of a signal-flow graph.

Frequency domain techniques A method of analyzing and designing linear control systems by using transfer functions and the Laplace transform as well as frequency response techniques.

Frequency response techniques A method of analyzing and designing control systems by using the sinusoidal frequency response characteristics of a system.

Gain The ratio of output to input; usually used to describe the amplification in the steady state of the magnitude of sinusoidal inputs, including dc.

Gain margin The amount of additional open-loop gain, expressed in decibels (dB), required at 180° of phase shift to make the closed-loop system unstable.

Gain-margin frequency The frequency at which the phase frequency response plot equals 180°. It is the frequency at which the gain margin is measured.

Homogeneous solution *See* **natural response.**

Ideal derivative compensator *See* **proportional-plus-derivative controller**.

Ideal integral compensator *See* **proportional-plus-integral controller**.

Instability The characteristic of a system defined by a natural response that grows without bounds as time approaches infinity.

Kirchhoff's law The sum of voltages around a closed loop equals zero. Also, the sum of currents at a node equals zero.

Lag compensator A transfer function, characterized by a pole on the negative real axis close to the origin and a zero close and to the left of the pole, that is used for the purpose of improving the steady-state error of a closed-loop system.

Lag-lead compensator A transfer function, characterized by a pole-zero configuration that is the combination of a lag and a lead compensator, that is used for the purpose of improving both the transient response and the steady-state error of a closed-loop system.

Laplace transformation A transformation that transforms linear differential equations into algebraic expressions. The transformation is especially useful for modeling, analyzing, and designing control systems as well as solving linear differential equations.

Lead compensator A transfer function, characterized by a zero on the negative real axis and a pole to the left of the zero, that is used for the purpose of improving the transient response of a closed-loop system.

Linear combination A linear combination of n variables, x_i, for $i = 1$ to n, given by the following sum, S:

$$S = K_n X_n + K_{n-1} X_{n-1} + \cdots + K_1 X_1$$

where each K_i is a constant.

Linear independence The variables x_i, for $i = 1$ to n, are said to be linearly independent if their linear combination, S, equals zero *only* if every $K_i = 0$ and *no* $x_i = 0$. Alternatively, if the x_i's are linearly independent, then $K_n x_n + K_{n-1} x_{n-1} + \cdots + K_1 x_1 = 0$ cannot be solved for any x_k. Thus, no x_k can be expressed as a linear combination of the other x_i's.

Linear system A system possessing the properties of superposition and homogeneity.

Linearization The process of approximating a nonlinear differential equation with a linear differential equation valid for small excursions about equilibrium.

Loop gain For a signal-flow graph, the product of branch gains found by traversing a path that starts at a node and ends at the same node without passing through any other node more than once, and following the direction of the signal flow.

Major-loop compensation A method of feedback compensation that adds a compensating zero to the open-loop transfer function for the purpose of improving the transient response of the closed-loop system.

Marginal stability The characteristic of a system defined by a natural response that neither decays nor grows, but remains constant or oscillates as time approaches infinity as long as the input is not of the same form as the system's natural response.

Mason's rule A formula from which the transfer function of a system consisting of the interconnection of multiple subsystems can be found.

Mechanical rotational impedance The ratio of the Laplace transform of the torque to the Laplace transform of the angular displacement.

Mechanical translational impedance The ratio of the Laplace transform of the force to the Laplace transform of the linear displacement.

Minor-loop compensation A method of feedback compensation that changes the poles of a forward-path transfer function for the purpose of improving the transient response of the closed-loop system.

Modern approach to control systems *See* **state-space representation.**

Natural frequency The frequency of oscillation of a system if all the damping is removed.

Natural response That part of the total response function due to the system and the way the system acquires or dissipates energy.

Negative feedback The case where a feedback signal is subtracted from a previous signal in the forward path.

Newton's law The sum of forces equals zero. Alternatively, after bringing the *ma* force to the other side of the equality, the sum of forces equals the product of mass and acceleration.

Nichols chart The locus of constant closed-loop magnitude and closed-loop phase frequency responses for unity feedback systems plotted on the open-loop dB versus phase-angle plane. It allows the closed-loop frequency response to be determined from the open-loop frequency response.

Nodes Points in a signal-flow diagram that represent signals.

No-load speed The speed produced by a motor with constant input voltage when the torque at the armature is reduced to zero.

Nonminimum-phase system A system whose transfer function has zeros in the right half-plane. The step response is characterized by an initial reversal in direction.

Nontouching-loop gain The product of loop gains from nontouching loops taken two, three, four, and so on at a time.

Nontouching loops Loops that do not have any nodes in common.

Notch filter A filter whose magnitude frequency response dips at a particular sinusoidal frequency. On the s-plane, it is characterized by a pair of complex zeros near the imaginary axis.

Nyquist criterion If a contour, A, that encircles the entire right half-plane is mapped through $G(s)H(s)$, then the number of closed-loop poles, Z, in the right half-plane equals the number of open-loop poles, P, that are in the right half-plane minus the number of counterclockwise revolutions, N, around -1, of the mapping; that is, $Z = P - N$. The mapping is called the *Nyquist diagram* of $G(s)H(s)$.

Nyquist diagram (plot) A polar frequency response plot made for the open-loop transfer function.

Nyquist sampling rate The minimum frequency at which an analog signal should be sampled for correct reconstruction. This frequency is twice the bandwidth of the analog signal.

Observability A property of a system by which an initial state vector, $x(t_0)$, can be found from $u(t)$ and $y(t)$ measured over a finite interval of time from t_0. Simply stated, observability is the property by which the state variables can be estimated from a knowledge of the input, $u(t)$, and output, $y(t)$.

Observer A system configuration from which inaccessible states can be estimated.

Octave Frequencies that are separated by a factor of two.

Ohm's law For dc circuits the ratio of voltage to current is a constant called resistance.

Open-loop system A system that does not monitor its output nor correct for disturbances.

Open-loop transfer function For a generic feedback system with $G(s)$ in the forward path and $H(s)$ in the feedback path, the open-loop transfer function is the product of the forward-path transfer function and the feedback transfer function, or $G(s)H(s)$.

Operational amplifier An amplifier—characterized by a very high input impedance, a very low output impedance, and a high gain—that can be used to implement the transfer function of a compensator.

Output equation For linear systems, the equation that expresses the output variables of a system as linear combinations of the state variables.

Overdamped response A step response of a second-order system that is characterized by no overshoot.

Partial-fraction expansion A mathematical equation where a fraction with n factors in its denominator is represented as the sum of simpler fractions.

Particular solution *See* **forced response.**

Passive network A physical network that only stores or dissipates energy. No energy is produced by the network.

Peak time, T_p The time required for the underdamped step response to reach the first, or maximum, peak.

Percent overshoot, *%OS* The amount that the underdamped step response overshoots the steady-state, or final, value at the peak time, expressed as a percentage of the steady-state value.

Phase margin The amount of additional open-loop phase shift required at unity gain to make the closed-loop system unstable.

Phase-margin frequency The frequency at which the magnitude frequency response plot equals zero dB. It is the frequency at which the phase margin is measured.

Phase variables State variables such that each subsequent state variable is the derivative of the previous state variable.

Phasor A rotating vector that represents a sinusoid of the form $A\cos(\omega t + \phi)$.

Pickoff point A block diagram symbol that shows the distribution of one signal to multiple subsystems.

Plant or process The subsystem whose output is being controlled by the system.

Poles (1) The values of the Laplace transform variable, s, that cause the transfer function to become infinite; and (2) any roots of factors of the characteristic equation in the denominator that are common to the numerator of the transfer function.

Position constant $\lim_{s\to 0} G(s)$

Positive feedback The case where a feedback signal is added to a previous signal in the forward path.

Proportional-plus-derivative (PD) controller A controller that feeds forward to the plant a proportion of the actuating signal plus its derivative for the purpose of improving the transient response of a closed-loop system.

Proportional-plus-integral (PI) controller A controller that feeds forward to the plant a proportion of the actuating signal plus its integral for the purpose of improving the steady-state error of a closed-loop system.

Proportional-plus-integral-plus-derivative (PID) controller A controller that feeds forward to the plant a proportion of the actuating signal plus its integral plus its derivative for the purpose of improving the transient response and steady-state error of a closed-loop system.

Quantization error For linear systems, the error associated with the digitizing of signals as a result of the finite difference between quantization levels.

Raible's tabular method A tabular method for determining the stability of digital systems that parallels the Routh-Hurwitz method for analog signals.

Rate gyro A device that responds to an angular position input with an output voltage proportional to angular velocity.

Residue The constants in the numerators of the terms in a partial-fraction expansion.

Rise time, T_r The time required for the step response to go from 0.1 of the final value to 0.9 of the final value.

Root locus The locus of closed-loop poles as a system parameter is varied. Typically, the parameter is gain. The locus is obtained from the open-loop poles and zeros.

Routh-Hurwitz criterion A method for determining how many roots of a polynomial in s are in the right half of the s-plane, the left half of the s-plane, and on the imaginary axis. Except in some special cases, the Routh-Hurwitz criterion does not yield the coordinates of the roots.

Sensitivity The fractional change in a system characteristic for a fractional change in a system parameter.

Settling time, T_s The amount of time required for the step response to reach and stay within $\pm 2\%$ of the steady-state value. Strictly speaking, this is the definition of

the 2% settling time. Other percentages, for example 5%, also can be used. This book uses the 2% settling time.

Signal-flow graph A representation of the interconnection of subsystems that form a system. It consists of nodes representing signals and lines representing subsystems.

Similarity transformation A transformation from one state-space representation to another state-space representation. Although the state variables are different, each representation is a valid description of the same system and the relationship between the input and the output.

Stability That characteristic of a system defined by a natural response that decays to zero as time approaches infinity.

Stall torque The torque produced at the armature when a motor's speed is reduced to zero under a condition of constant input voltage.

State equations A set of n simultaneous, first-order differential equations with n variables, where the n variables to be solved are the state variables.

State space The n-dimensional space whose axes are the state variables.

State-space representation A mathematical model for a system that consists of simultaneous, first-order differential equations and an output equation.

State-transition matrix The matrix that performs a transformation on $\mathbf{x}(0)$, taking $\mathbf{x}$ from the initial state, $\mathbf{x}(0)$, to the state $\mathbf{x}(t)$ at any time, $t \geq 0$.

State variables The smallest set of linearly independent system variables such that the values of the members of the set at time t_0 along with known forcing functions completely determine the value of all system variables for all $t \geq t_0$.

State vector A vector whose elements are the state variables.

Static error constants The collection of position constant, velocity constant, and acceleration constant.

Steady-state error The difference between the input and the output of a system after the natural response has decayed to zero.

Steady-state response *See* **forced response**.

Subsystem A system that is a portion of a larger system.

Summing junction A block diagram symbol that shows the algebraic summation of two or more signals.

System type The number of pure integrations in the forward path of a unity feedback system.

System variables Any variable that responds to an input or initial conditions in a system.

Tachometer A voltage generator that yields a voltage output proportional to rotational input speed.

Time constant The time for e^{-at} to decay to 37% of its original value at $t = 0$.

Time-domain representation *See* **state-space representation**.

Torque-speed curve The plot that relates a motor's torque to its speed at a constant input voltage.

Transducer A device that converts a signal from one form to another, for example from a mechanical displacement to an electrical voltage.

Transfer function The ratio of the Laplace transform of the output of a system to the Laplace transform of the input.

Transient response That part of the response curve due to the system and the way the system acquires or dissipates energy. In stable systems it is the part of the response plot prior to the steady-state response.

Tustin transformation A bilinear transformation that converts transfer functions from continuous to sampled and vice versa. The important characteristic of the Tustin transformation is that both transfer functions yield the same output response at the sampling instants.

Type *See* **system type**.

Undamped response The step response of a second-order system that is characterized by a pure oscillation.

Underdamped response The step response of a second-order system that is characterized by overshoot.

Velocity constant $\lim_{s \to 0} sG(s)$

***z*-transformation** A transformation related to the Laplace transformation that is used for the representation, analysis, and design of sampled signals and systems.

Zero-input response That part of the response that depends upon only the initial state vector and not the input.

Zero-order sample-and-hold (z.o.h.) A device that yields a staircase approximation to the analog signal.

Zeros (1) Those values of the Laplace transform variable, s, that cause the transfer function to become zero; and (2) any roots of factors of the numerator that are common to the characteristic equation in the denominator of the transfer function.

Zero-state response That part of the response that depends upon only the input and not the initial state vector.

This page intentionally left blank

Answers to Selected Problems

Chapter 1

19. c. $x(t) = \frac{2}{5} - e^{-4t}\left(\frac{2}{5}\cos 3t + \frac{8}{15}\sin 3t\right)$

20. b. $x(t) = -e^{-t} + 9te^{-t} + 5e^{-2t} + t - 2$

Chapter 2

3. b. $x(t) = \frac{15}{26}e^{-2t} - \frac{3}{10}e^{-4t} - \frac{18}{65}\cos 3t - \frac{1}{65}\sin 3t$

7. $\frac{Y(s)}{X(s)} = \frac{s^3 + 4s^2 + 6s + 8}{s^3 + 3s^2 + 5s + 1}$

8. c. $\frac{d^3x}{dt^3} + 11\frac{d^2x}{dt^2} + 12\frac{dx}{dt} + 18x = \frac{df}{dt} + 3f(t)$

16. a. $\frac{V_o(s)}{V_i(s)} = \frac{1}{s+2}$

18. b. $\frac{V_o(s)}{V_i(s)} = \frac{s^2 + 2s + 2}{s^4 + 2s^3 + 3s^2 + 3s + 2}$

33. $\frac{\theta_2(s)}{T(s)} = \frac{3}{20s^2 + 13s + 4}$

34. $\frac{\theta_2(s)}{T(s)} = \frac{1}{130s^2 + 400s + 55}$

43. $\frac{\theta_2(s)}{E_a(s)} = \frac{0.0833}{s(s + 0.75)}$

Chapter 3

1. $\dot{\mathbf{x}} = \begin{bmatrix} -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\ -\frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} i_{L_1} \\ i_{L_2} \\ v_o \end{bmatrix} + \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} v_i$

$$y = [0 \quad 0 \quad 1] \begin{bmatrix} i_{L_1} \\ i_{L_2} \\ v_o \end{bmatrix}$$

Note: L_1 is left-most inductor in Figure P3.1 in the text.

11. a. $\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -13 & -5 & -1 & -5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r(t)$

$$c(t) = [10 \quad 8 \quad 0 \quad 0]\mathbf{x}$$

14. a. $\dfrac{Y(s)}{R(s)} = \dfrac{10}{s^3 + 5s^2 + 2s + 3}$

19. $\dot{\mathbf{x}} = \begin{bmatrix} -\frac{D_{eq}}{J_{eq}} & 0 & \frac{K_t N_1}{J_{eq} N_2} \\ 1 & 0 & 0 \\ -\frac{K_b N_2}{L_a N_1} & 0 & -\frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} \omega_L \\ \theta_L \\ i_a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} e_a$

$$y = \begin{bmatrix} 0 & \frac{N_2}{N_1} & 0 \end{bmatrix} \begin{bmatrix} \omega_L \\ \theta_L \\ i_a \end{bmatrix}$$

Chapter 4

14. $x(t) = \frac{1}{5}\left[1 - \sqrt{\frac{20}{19}} e^{-0.5t} \cos\left(\frac{\sqrt{19}}{2}t - \arctan\frac{1}{\sqrt{19}}\right)\right]$

20. a. $\zeta = 0.375$; $\omega_n = 4$ rad/s; $T_s = 2.67$ s; $T_p = 0.847$ s; $\%OS = 28.06$

23. a. $s = -6.67 \pm j9.88$

35. $s = -5.79, -1.21$

36. a. $s^3 - 8s^2 - 11s + 8 = 0$ **b.** $s = 9.11, 0.534, -1.64$

40. $y(t) = \frac{2}{5} - \frac{2}{5}e^{-5t}$

43.

$$\mathbf{\Phi}(t) = \begin{bmatrix} 1.0455e^{-0.20871t} - 0.045545e^{-4.7913t} & 0.21822e^{-0.20871t} - 0.21822e^{-4.7913t} \\ -0.21822e^{-0.20871t} + 0.21822e^{-4.7913t} & -0.045545e^{-0.20871t} + 1.0455e^{-4.7913t} \end{bmatrix}$$

$$\mathbf{x}(t) = \begin{bmatrix} 1.0455e^{-0.20871t} - 0.045545e^{-4.7913t} \\ -0.21822e^{-0.20871t} + 0.21822e^{-4.7913t} \end{bmatrix}$$

$$y(t) = 0.60911e^{-0.20871t} + 0.39089e^{-4.7913t}$$

73. $D = 0.143$ N-m-s/rad

76. $R = 912\,\Omega$

Chapter 5

2. $\dfrac{C(s)}{R(s)} = \dfrac{G_3(G_1G_2 + 1)}{1 + G_1H_1}$

4. $\dfrac{C(s)}{R(s)} = \dfrac{G_1G_2 + G_3}{1 + G_3H + G_1G_2H + G_2G_4}$

6. $\dfrac{C(s)}{R(s)} = \dfrac{G_1G_5}{1 + G_1G_2 + G_1G_3G_4G_5 + G_1G_3G_5G_6G_7 + G_1G_5G_8}$

9. $\dfrac{C(s)}{R(s)} = \dfrac{G_4G_6 + G_2G_5G_6 + G_3G_5G_6}{1 + G_6 + G_1G_2 + G_1G_3 + G_1G_2G_6 + G_1G_3G_6 + G_4G_6G_7 + G_2G_5G_6G_7 + G_3G_5G_6G_7}$

26. $\dfrac{C(s)}{R(s)} = \dfrac{G_1G_2G_3G_4}{2 + G_2G_3G_4 + 2G_3G_4 + 2G_4}$

27. $\dfrac{C(s)}{R(s)} = \dfrac{G_1G_6G_7(G_2 + G_3)(G_4 + G_5)}{1 - G_6G_7H_3(G_2 + G_3)(G_4 + G_5) - G_6H_1 - G_7H_2 + G_6G_7H_1H_2}$

28. $\dfrac{C(s)}{R(s)} = \dfrac{s^3 + 1}{2s^4 + s^2 + 2s}$

30. b. $\dot{\mathbf{x}} = \begin{bmatrix} -5 & 1 & 0 & 0 \\ 0 & -5 & 0 & 0 \\ 0 & 0 & -7 & 1 \\ 0 & 0 & 0 & -7 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} r(t)$

$$y = \begin{bmatrix} -\frac{3}{4} & 1 & -\frac{5}{4} & -1 \end{bmatrix} \mathbf{x}$$

37. $\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r(t)$

$c = [-1 \quad 1 \quad 0 \quad 0]\mathbf{x}$

73. $D_L = 3560$ N-m-s/rad

Chapter 6

1. 2 rhp, 3 lhp, 0 $j\omega$

3. 3 rhp, 2 lhp, 0 $j\omega$

4. 1 rhp, 0 lhp, 4 $j\omega$
5. 0 rhp, 2 lhp, 2 $j\omega$
9. Unstable
17. 1 rhp, 2 lhp, 4 $j\omega$
23. $K > \frac{3}{4}$; $K < -1$
39. **a.** $-4 < K < 20.41$; **b.** 1.36 rad/s
41. **a.** $0 < K < 19.69$; **b.** $K = 19.69$; **c.** $s = \pm\, j1.118,\ -4.5,\ -3.5$
42. $-\frac{2}{3} < K < 0$

Chapter 7

4. $e_{\text{step}}(\infty) = 0$; $e_{\text{ramp}}(\infty) = 127.5$; $e_{\text{parabola}}(\infty) = \infty$
7. $\dot{e}(\infty) = 0.9$
10. **a.** $\%OS = 14.01$; **b.** $T_s = 0.107$ sec; **c.** $e_{\text{step}}(\infty) = 0$;
d. $e_{\text{ramp}}(\infty) = 0.075$; **e.** $e_{\text{parabola}}(\infty) = \infty$
13. **a.** $K_p = \frac{1}{3}$, $K_v = 0$, $K_a = 0$; **b.** $e(\infty) = 37.5,\ \infty,\ \infty,$ respectively;
c. Type 0
20. $K = 110,000$
27. $\beta = 1$, $K = 1.16$, $\alpha = 7.76$, or $\beta = -1$, $K = 5.16$, $\alpha = 1.74$
31. **a.** $K = 831,744$, $a = 831.744$
35. $K_1 = 125,000$, $K_2 = 0.016$
41. **a.** Step: $e(\infty) = 1.098$; ramp: $e(\infty) = \infty$

Chapter 8

16. Breakaway point $= -2.333$; asymptotes: $\sigma_a = -5$; $j\omega$-axis crossing $= \pm\, j\,7.35$
19. **b.** Asymptotes: $\sigma_a = -\frac{8}{3}$; **c.** $K = 140.8$; **d.** $K = 13.12$
20. $K = 9997$; $\alpha = 7$
23. **a.** $\sigma_a = -\frac{5}{2}$; **b.** $s = -1.38,\ -3.62$; **c.** $0 < K < 126$; **d.** $K = 10.3$
26. **b.** $K = 9.4$; **c.** $T_s = 4.62$ s, $T_p = 1.86$ s; **d.** $s = -4.27$; **e.** $0 < K < 60$
30. $\alpha = 9$
39. **a.** $0 < K < 4366$; **b.** $K = 827.2$; **c.** $K = 527.6$
42. **a.** $K = 170.1$; **b.** $K = 16.95$

Chapter 9

1. $G_c(s) = \frac{s + 0.1}{s}$; $K \simeq 72.23$ for both cases; $K_{p_O} = 2.44$; $K_{p_N} = \infty$;
$\%OS_O = \%OS_N = 16.3$; $T_{s_O} = T_{s_N} = 2.65$ s

9. **a.** $s = -2.5 \pm j5.67$; **b.** Angle $= -59.27°$; **c.** $s = -5.87$
d. $K = 225.79$; **e.** $s = -11.59, -1.36$

10. **a.** $s = -2.4 \pm j4.16$; **b.** $s = -6.06$; **c.** $K = 29.12$;
d. $s = -1.263$; **f.** $K_a = 4.8$

14. **a.** $G_c(s) = \dfrac{s+7}{s+37.42}$, $K = 5452$; dominant poles $= -4.13 \pm j10.78$

24. **a.** $K_{uc} = 10$; $K_c = 9.95$; **b.** $K_{p_{uc}} = 1.25$; $K_{p_c} = 6.22$;
c. $\%OS_{uc} = \%OS_c = 4.32$;
d. Uncompensated: exact second-order system, approximation OK; compensated: closed-loop pole at -0.3, closed-loop zero at -0.5, simulate
e. Approach to final value longer than settling time of uncompensated system
f. $G_{\text{LLC}}(s) = \dfrac{404.1(s+0.5)(s+4)}{(s+2)(s+4)(s+0.1)(s+28.36)}$ yields approximately a 5 times improvement in speed.

25. $G_c(s) = \dfrac{(s+6.93)(s+0.1)}{s}$, $K = 3.08$

28. Poles $= -0.747 \pm j1.237, -2.51$; zeros—none

Chapter 10

10. System 1: $0 < K < 490.2$; System 2: $0 < K < 1.4$; System 3: $1 < K < \infty$ (Answers are from exact frequency response)

11. **a.** System 1: $G_M = -6.38$ dB; $\Phi_M = -20.3°$ (Answers are from exact frequency response)

15. **c.** $\omega_{\text{BW}} = 2.29$ rad/s

23. System 2: $T_s = 2.23$ sec, $T_p = 0.476$ s, $\%OS = 42.62$ (Answers are from exact frequency response)

44. $G_M = 1.17$ dB, $\Phi_M = 6.01°$ (Answers are from exact frequency response)

Chapter 11

1. **a.** $K = 2113$ (Answer is from exact frequency response)

2. **a.** $K = 2365$ (Answer is from exact frequency response)

3. **a.** $K = 575$ (Answer is from exact frequency response)

12. $G_c(s) = \dfrac{s+2.701}{s+5.954}$, $K = 7936$

(Answer is from exact frequency response)

21. $G_c(s) = \dfrac{(s+0.097)(s+2.16)}{s}$, $K = 25.189$

(Answer is from exact frequency response)

Chapter 12

1. **d.** For function **i**: $T(s) = \dfrac{s+3}{s^2 + (k_2+8)s + (k_1+16)}$

3. b. For function **i**: $G(s) = \dfrac{6.25}{s} - \dfrac{27.5}{s+10} + \dfrac{71.25}{s+20}, \quad T(s) = \dfrac{200(s^2+7s+25)}{4s^3+as^2+bs+c}$

where $a = (25k_3 - 110k_2 + 285k_1 + 120)$
$b = (750k_3 - 2200k_2 + 2850k_1 + 800)$
$c = 5000k_3$
and $\mathbf{C} = [1 \quad 1 \quad 1]; \quad \mathbf{B} = [71.25 \quad -27.5 \quad 6.25]^T$ was used

11. a. Uncontrollable; **b.** Controllable; **c.** Controllable

14. $\mathbf{K} = [92.35 \quad 36.78 \quad -7]$ for a characteristic polynomial of $(s+6)(s^2+8s+45.78) = s^3+14s^2+93.78s+274.7$

23. $\mathbf{L} = [-671.19 \quad 1472.4]^T$ for a characteristic polynomial of $s^2+144s+14,400$

Chapter 13

3. a. $f(kT) = 229.5(0.4)^k - 504(0.6)^k + 275.5(0.8)^k$

6. c. $G(z) = 0.395\,\dfrac{(z+0.2231)}{(z-0.2231)(z^2+0.1857z+0.04979)}$

8. b. $G(z) = 0.0517\,\dfrac{z^2+2.2699z+0.2995}{(z-1)(z-0.2231)(z-0.4065)}$

9. a. $T(z) = \dfrac{G_1(z)G_2(z)}{1+G_1(z)G_2H(z)}$

14. $0 < K < 15.76$

15. a. $K_p = \dfrac{1}{2},\ e^*(\infty) = \dfrac{2}{3}; \quad K_v = 0,\ e^*(\infty) = \infty; \quad K_a = 0,\ e^*(\infty) = \infty$

17. $K = 14.18$ for 16.3% of overshoot; $0 < K < 109.28$ for stability

Credits

Figure and Photo Credits

Figures, photos, Case Studies, Examples, and Problems in Chapters 4, 5, 6, 8, 9, 13, Appendix B, and rear endpapers: Adapted from Johnson, H., et al. *Unmanned Free-Swimming Submersible (UFSS) System Description*, NRL Memorandum Report 4393 (Washington, D.C.: Naval Research Laboratory, 1980). MATLAB screen shots in Appendixes C and E were reprinted with permission from The MathWorks.

Chapter 1

1.3: (**a**): © Bettman/Corbis; (**b**): ART on FILE/© Corbis. **1.4:** © Hank Morgan/Rainbow/PNI. **1.5b:** Pioneer Electronics (USA), Inc. **1.7:** © Donald Swartz/iStockphoto. **1.8:** © Peter Menzel. **P1.3:** Adapted from Ayers, J. Taking the Mystery Out of Winder Controls, *Motion System Design*, April 1988. Penton Media, Inc. **P1.5:** Jenkins, H. E.; Kurfess, T. R.; and Ludwick, S. J. Determination of a Dynamic Grinding Model, *Journal of Dynamic Systems, Measurements and Control*, vol. 119, June 1997, p. 290. 1997 ASME. Reprinted with permission of ASME. **P1.6:** Vaughan, N. D., and Gamble, J. B. The Modeling and Simulation of a Proportional Solenoid Valve, *Journal of Dynamic Systems, Measurements and Control*, vol. 118, March 1996, p. 121. 1996 ASME. Reprinted with permission of ASME. **P1.7(a), (b), (c):** Reprinted figure with permission from Bechhofer, J., *Feedback for Physicists: A Tutuorial Essay on Control, Reviews of Modern Physics*, pp. 77, 783, 2005. Copyright (2007) by the American Physical Society. **P1.8:** D. A. Weinstein/Custom Medical Stock Photo. **P1.11:** O'Connor, D. N.; Eppinger, S. D.; Seering, W. P.; and Wormly, D. N. Active Control of a High-Speed Pantograph, *Journal of Dynamic Systems, Measurements and Control*, vol. 119, March 1997, p. 2. 1997 ASME. Reprinted with permission of ASME. **P1.12, P1.13:** Mark Looper, www.Altfuels.org.; **P1.14:** Robert Bosch GmbH, *Bosch Automotive Handbook,* 7th ed., Wiley UK.

Chapter 2

2.34: © Debra Lex. **2.51:** Adapted from Milsum, J. H. *Biological Control Systems Analysis* (New York: McGraw-Hill, 1966), p. 182. © 1966 McGraw-Hill, Inc. Used with permission of the publisher. **P2.36:** Lin Jung-Shan, Kanellakopoulos Ioannis. Nonlinear Design of Active Suspensions, *IEEE Control Systems Magazine*, vol. 17, no. 3. June 1997, pp. 45–49. Fig. 1, p. 46. **P2.37:** Marttinen A., Virkkunen J., Salminen R. T. Control Study with Pilot Crane, *IEEE Trans. on Education*, vol. 33, No. 3, August 1990. Fig. 2, p. 300. **P2.38:** Wang J. Z., Tie B., Welkowitz W., Semmlow J. L., Kostis J. B. Modeling Sound Generation in Stenosed Coronary Arteries, *IEEE Trans. on Biomedical Engineering*, vol. 37, no. 11, November 1990. **P2.39(a):** From O'Connor, D. N.; Eppinger, S. D.; Seering, W. P.; and Wormly, D. N. Active Control of a High-Speed Pantograph, *Journal of Dynamic Systems, Measurements and Control*, vol. 119, March 1997, p. 2. 1997 ASME. Reprinted with permission of ASME. **P2.39(b):** Adapted from O'Connor, D. N.; Eppinger, S. D.; Seering, W. P.; and Wormly, D. N. Active Control of a High-Speed Pantograph, *Journal of Dynamic Systems, Measurement and Control*, vol. 119, March 1997, p. 3. 1997 ASME. Reprinted with permission of ASME. **P2.40:** Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Trans. on Education*, vol. 47, no. 1, February 2004, pp. 65–73. **P2.41:** Robert Bosch GmbH, *Bosch Automotive Handbook*, 7th ed., Wiley UK.

Chapter 3

3.13: Bruce Frisch/S.S./Photo Researchers. **P3.14(a), (b):** Hong, J.; Tan, X.; Pinette, B.; Weiss, R.; and Riseman, E. M. Image-Based Homing, *IEEE Control Systems*, Feb. 1992, pp. 38–45. © 1992 IEEE. **P3.15(a), (b):** Adapted from Cavallo, A.; De Maria, G.; and Verde, L. Robust Flight Control Systems: A Parameter Space Design, *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, September–October, 1992, pp. 1210–1211. 1992 AIAA. Reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc. **P3.16:** Adapted from Chiu, D. K., and Lee, S. Design and Experimentation of a Jump Impact Controller, *IEEE Control Systems*, June 1997, Fig. 1, p. 99. 1997 IEEE. Reprinted with permission. **P3.17:** Liceaga-Castro, E., vander Molen G. M. Submarine H^{∞} Depth Control Under Wave Disturbances, *IEEE Trans. on Control Systems Technology,* vol. 3, no. 3, 1995. Fig. 1, p. 339. **P3.18:** Li, S., Jarvis, A. J., Leedal, D. T., Are Response Function Representations of the Global Carbon Cycle Ever Interpretable?, Tellus vol. 61B, Wiley UK. **P3.19:** Preitl, Z., Bauer, P., Bokor, J., A Simple Control Solution for Traction Motor Used in Hybrid Vehicles, 4th International Symposium on Applied Computational intelligence and Informantics, 2007. 4th International Symposium on Applied Computational Intelligence and Informatics, IEEE, 2007, Fig. 8, p. 2. **P3.31(c):** (table) Craig, I. K., Xia, X., and Venter, J. W. Introducing HIV/AIDS Education

into the Electrical Engineering Curriculum at the University of Pretoria. *IEEE Trans. on Education*, vol. 47, no. 1, February 2004, pp. 65–73. Table II, p. 67.

Chapter 4

4.22: Yoshikazu Tsuno/AFP/Getty Images, Inc. **4.27:** Adapted from Dorf, R. C. *Introduction to Electric Circuits*, 2nd ed. (New York: John Wiley & Sons, 1989, 1993), p. 583. © 1989, 1993 John Wiley & Sons. Reprinted by permission of the publisher. **4.33:** Courtesy of Naval Research Laboratory. **P4.11:** Courtesy of Pacific Robotics, Inc. **P4.12:** Borovic B., Liu A. Q., Popa D., Lewis F. L. *Open-loop versus closed-loop control of MEMS devices: choices and issues.* J. Micromech. Microeng. vol. 15, 2005. Fig. 4, p. 1919. **P4.14:** DiBona G. F. *Physiology in Perspective: The Wisdom of the Body. Neural Control of the Kidney*, Am. J. Physiol. Regul. Integr. Comp. Physiol. vol. 289, 2005. Fig. 6, p. R639. Used with permission. **P4.24:** From Manring, N. D., and Johnson, R. E. Modeling and Designing a Variable Displacement Open-Loop Pump, *Journal of Dynamic Systems, Measurement and Control*, vol. 118, June 1996, p. 268. 1996 ASME. Reprinted with permission of ASME.

Chapter 5

5.1: NASA-Houston. **5.33:** © Rob Catanach, Woods Hole Oceanographic Institution. **P5.32:** Tanis, D. *Space Shuttle GN&C Operations Manual* (Downey, CA: Rockwell International), August 1988. **P5.35(a):** Courtesy of Hank Morgan/Rainbow/ PNI. **P5.36(a):** Bailey, F. N.; Cockburn, J. C.; and Dee, A. Robust Control for High-Performance Materials Testing, *IEEE Control Systems*, April 1992, p. 63. © 1992 IEEE. **P5.38:** Lepschy A. M., Mian G. A., Viaro U. Feedback Control in Ancient Water and Mechanical Clocks, *IEEE Trans. on Education,* vol. 35, 1992. Figs. 1 and 2, p. 4. **P5.39:** Ben-Dov D., Salcudean S. E. A Force-Controlled Pneumatic Actuator, *IEEE Trans. on Robotics and Automation*, vol. 11, 1995. Fig. 6, p. 909. **P5.43:** Lin Jung-Shan, Kanellakopoulos Ioannis. Nonlinear Design of Active Suspensions, *IEEE Control Systems Magazine*, vol. 17, no. 3. June 1997, pp. 45–49. Fig. 3, p. 48. **P5.44:** de Vlugt, Schouten A. C., van der Helm F. C. T. *Adaptation of reflexive feedback during arm posture to different environments*, Biol. Cybern. vol. 87, 2002. Fig. A1, p. 24. by **P5.45:** Karkoub, M., Her, M-G., Chen, J.M., Design and Control of a Haptic Interactive Motion, Robotica vol. 28, 2008. Copyright © 2010 Cambridge University Press. Reprinted with permission. **P5.46:** Piccin, O., Barbe, L., Bayle, B., de Mathelin, M., Gangi, A., A Force Teleoperated Needle Insertion Device Percutaneous Procedures, International Journal of Robotics Research, vol. 28, 2009. Reprinted with permission from Sage Publications. **P5.47:** Lee, S.R., Ko, S., Dehbonei, H., Jeon, C.H., Kwon, O.S., Operational Characteristics of PV/Diesel Hybrid Distributed Generation System Using Dual Voltage Source Inverter for Weak Grid, ISIS 2007 Proceedings, 8th Symposium on Advanced Int. Sys. 2007. **P5.48:** Kong, F., de Keyser, R., Identification and Control of the Mold Level in a Continuous Casting Machine, 2nd IEEE Conference on Control Applications. Vancouver, B.C., 1993. Fig. 1, p. 53, pp. 53–58.

Chapter 6

6.9: Courtesy of Woods Hole Oceanographic Institution. **6.11:** Courtesy of FANUC Robotics North America, Inc. **P6.14(a):** Courtesy of Kazuhiko Kawamura, Vanderbilt University. **P6.16:** From Rober, S. J.; Shin, Y. C.; and Nwokah, O. D. I. A Digital Robust Controller for Cutting Force Control in the End Milling Process, *Journal of Dynamic Systems, Measurement and Control*, vol. 119, June 1997, p. 147. 1997 ASME. Reprinted with permission of ASME. **P6.17(a):** © Japan Air Lines/ Photo Researchers. **P6.17(b):** Adapted from Bittar, A., and Sales, R. M. H_2 and H_∞ Control for MagLev Vehicles, *IEEE Control Systems,* vol. 18, no. 4, August 1998, Equations 7, 8, and Table 2 on pp. 20–21. © 1998 IEEE. Reprinted with permission.

Chapter 7

7.9: Chuck O'Rear/Westlight/Corbis Images. **7.23(a):** Isailovic, J., *Videodisc and Optical Memory Technologies*, © 1985, p. 77. Reprinted by permission of Pearson Education, Inc., Upper Saddle River, N.J. **P7.22:** Lam C. S., Wong M. C., Han Y. D. *Stability Study on Dynamic Voltage Restorer (DVR)* Power Electronics Systems and Applications 2004, Proceedings First International Conference on Power Electronics 2004. Fig. 7, p. 68. **P7.28(a), (c):** From Kumar, R. R.; Cooper, P. A.; and Lim, T. W. Sensitivity of Space Station Alpha Joint Robust Controller to Structural Modal Parameter Variations, *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 6, Nov/Dec 1992, pp. 1427–1428. © 1992 AIAA. Reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc. **P7.29:** Yin, G., Chen, N., Li, P., Improving Handling Stability Performance of Four-Wheel Steering Vehicle via μ-Synthesis Robust Control, IEEE Transactions on Vehicular Technology vol. 56, no. 5, 2007. Fig. 2, p. 2434. pp. 2432–2439 © 2007 IEEE. Reprinted with permission. **P7.32:** Hess, R. A.; Malsbury, T.; and Atencio, A., Jr. Flight Simulator Fidelity Assessment in a Rotorcraft Lateral Translation Maneuver, *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1 Jan/Feb 1993, p. 80. © 1992 AIAA. Reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc. **P7.33(a), (b):** Ohnishi, K.; Shibata, M.; and Murakami, T. Motion Control for Advanced Mechatronics, *IEEE/ASME Transactions on Mechatronics*, vol. 1, no. 1, March 1996, **(a):** Fig. 14, p. 62, **(b):** Fig. 16, p. 62. © 1996 IEEE. Reprinted with permission.

Chapter 8

8.4(a): Largeformat 4×5/iStockphoto. **P8.13:** Tony Dejak/©AP/Wide World Photos. North America, Inc. **P8.13(b):** Adapted from Hardy, H. L. Multi-Loop Servo Controls Programmed Robot, *Instruments and Control Systems*, June 1967, p. 105. **P8.14:** *GNC FSSR FC Ascent*, vol. 1, June 30, 1985 (Downey,

CA: Rockwell International). **P8.15(a):** Bruner, A. M.; Belvin, W. K.; Horta, L. G.; and Juang, J. Active Vibration Absorber for the CSI Evolutionary Model: Design and Experimental Results, *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, Sept/Oct 1992, p. 1254. © 1992 AIAA. Reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc. **P8.17:** Karlsson, P., and Svensson, J. DC Bus Voltage Control for a Distributed Power System, IEEE Trans. Power Electronics, vol. 18, no. 6, 2003. Fig. 4, p. 1406. pp. 1405–1412. (c) 2003 IEEE. **P8.20:** Cho, D., and Hedrick, J. K. Pneumatic Actuators for Vehicle Active Suspension Applications, *ASME Journal of Dynamic Systems, Measurement and Control*, March 1985, p. 68, Fig. 4. Reprinted with permission of ASME. **P8.23(a), (b):** Adapted from Annaswamy, A. M., and Ghonien, A. F. Active Control in Combustion Systems, *IEEE Control Systems*, December 1995, p. 50, 51, and 59. © 1995 IEEE. Reprinted with permission. **P8.24(a):** © Hammondovi/iStockphoto. **P8.24(b), (c):** Adapted from Anderson, C. G.; Richon, J.-B.; and Campbell, T. J. An Aerodynamic Moment-Controlled Surface for Gust Load Alleviation on Wind Turbine Rotors, *IEEE Transactions on Control System Technology*, vol. 6, no. 5, September 1998, pp. 577–595. © 1998 IEEE. **P8.25:** Reproduced by permission of M. Spong et al., Robot Modeling and Control. John Wiley & Sons. Hoboken, NJ, 2006. Fig. 6.20, p. 221. Reprinted with permission of John Wiley & Sons, Inc. **P8.26:** Reproduced by permission of M. Spong et al., Robot Modeling and Control. John Wiley & Sons. Hoboken, NJ, 2006. Fig 6.22, pg 223.

Chapter 9

9.46: Photo by Mark E. Van Dusen. **P9.5:** Romagnoli, J. A., and Palazoglu, A. *Introduction to Process Control,* CRC Press, Boca Raton, 2006. p. 44, Fig. 3.4. **P9.6:** Smith, C. A. *Automated Continuous Process Control.* John Wiley & Sons, New York, NY, 2002. p. 128, Fig. 6–1.1. **P9.14(a):** Cho, D.; Kato, Y.; and Spilman, D. Sliding Mode and Classical Controllers in Magnetic Levitation Systems, *IEEE Control Systems*, Feb. 1993, p. 43, Fig. 1. © 1993 IEEE. Reprinted with permission.

Chapter 10

10.1: Courtesy National Instruments Corporation © 2010 **P10.14:** © Michael Rosenfield/ Science Faction/©Corbis. **P10.15(a):** © Stephen Sweet/iStockphoto. **P10.17:** Van der Zalm, G., Huisman, G., Steinbuch, M., Veldpaus, F., Frequency Doman Approach for the Design of Heavy-Duty Vehicle Speed Controllers, International Journal of Heavy Vehicle Systems vol 15, no. 1.

Chapter 11

11.10(a), (b): Katharina Bosse/laif/Redux Pictures. **P11.3(a):** Rebecca Cook/Rueters/©Corbis.

Chapter 12

12.1: Robin Nelson/Zuma Press. **P12.6:** Tadeo F., Pérez, Loépez O., and Alvarez T. Control of Neutralization Processes by Robust Loopsharing. *IEEE Trans. on Cont. Syst. Tech.,* vol. 8, no. 2, 2000. Fig. 2, p. 239.

Chapter 13

13.12: © David J. Green—Industry/Alamy **13.28:** Adapted from Chassaing, R. *Digital Signal Processing* (New York: John Wiley & Sons, Inc, 1999), p. 137. © 1999 John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc. **13.29:** Adapted from Chassaing, R. *Digital Signal Processing* (New York: John Wiley & Sons, Inc., 1999), p. 137. © 1999 John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc. **13.36:** Adapted from Chassaing, R. *Digital Signal Processing* (New York: John Wiley & Sons, Inc, 1999), p. 137. © 1999 John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.

Figure Caption Sources

Back Cover: cover photo caption from Dennis Hong/Virginia Polytechnic Institute. **1.3(b):** The World of Otis, 1991, p. 2, and Tell Me About Elevators, 1991, pp. 20–25, United Technologies Otis Elevator Company. © 1991 Otis Elevator Company. **1.8:** Overbye, D. The Big Ear, *OMNI*, Dec. 1990, pp. 41–48. **5.33:** Ballard, R. D. *The Discovery of the Titanic* (New York: Warner Books, Inc., 1987). **6.9:** Ballard, R. D. The Riddle of the Lusitania, *National Geographic*, April 1994, pp. 68–85. **6.11:** FANUC Robotics North America, Inc., **7.9:** Bylinski, G. *Silicon Valley High Tech: Window to the Future* (Hong Kong: Intercontinental Publishing Corp., Ltd., 1985). **10.1:** National Instruments Corporation

Trademarks

AMTRAK is a registered trademark of National Railroad Passenger Corporation. **Adobe** and **Acrobat** are trademarks of Adobe Systems, Inc. which may be registered in some jurisdictions. **FANUC** is a registered trademark of FANUC, Ltd. **Microsoft, Visual Basic,** and **PowerPoint** are registered trademarks of Microsoft Corporation. **QuickBasic** is a trademark of Microsoft Corporation. **MATLAB** and **SIMULINK** are registered trademarks of The MathWorks, Inc. **The Control System Toolbox, LTI Viewer, Root Locus Design GUI, Symbolic Math Toolbox,** and **MathWorks** are trademarks of the MathWorks, Inc. **Intelligent Soft-Arm Control (ISAC)** is a trademark of the Intelligent Robotics Lab, Vanderbilt University. **HelpMate** is a trade-mark of HelpMate Robotics, Inc. **LabVIEW** is a registered trademark of National Instruments Corporation. **Segway** is a registered trademark of Segway, Inc. in the United States and/or other countries. **Chevrolet Volt** is a trademark of General Motors LLC. Virtual plant simulations pictured and referred to herein are trademarks or registered trademarks of Quanser Inc. and/or its affiliates. © 2010 Quanser Inc. All rights reserved. Quanser virtual plant simulations pictured and referred to herein may be subject to change without notice.

This page intentionally left blank

Index

Abscissa of convergence, 35n
Absorption, 146–148
Acceleration constant, 595
Achouayb, K., 115
Ackerman's formula, 678
Acoustic waves, 444
Active-circuit realization, of compensation, 504
Active queue management (AQM) algorithm, 383
Active suspension system, 291, 379, 443
Active vibration absorber, 440
Actuating signal, 9
Actuator block diagram, 290
Admittance, 54
AFTI/F-16 aircraft
  attack control system, 718
  bearing angle control, 527
Aggarwal, J. K., 115
AIDS. *See* HIV/AIDS
Aircraft attitude, 24
Aircraft longitudinal dynamics, 153
Aircraft roll control system, 658
Alertness monitoring, 27
Alvarez, T., 721
Alvin, 272
Amplifier saturation, 88
  load angular velocity response and, 197
Amplifier saturation simulation, with LabVIEW program, 878–880
Amplifiers
  operational, 58–61
  transfer functions and, 94
Amplitudes, 164
AMTRAK train active suspension system, 443
Anagnost, J. J., 337
Analogs
  explanation of, 84–85
  parallel, 86–87
  series, 85–86
Analog-to-digital control conversion, 726–727
  antenna azimuth for, 724
  steps to, 727
Analysis
  of control systems, 10–15, 18–19
  via input substitution, 366–367
Anderson, C. G., 447, 452
Anesthesia regulation, 26, 221
Angles of departure/arrival
  from complex pole, 409
  in root locus sketching, 407–409
Angular displacement, in lossless gears, 75
Angular velocity, 82
Annaswamy, A. M., 445, 452
Antenna azimuth, 12
  analog- to-digital control conversion for, 724
  block diagram for, 18
  position control system, 13–15, 208
Antenna azimuth position control, state-space design for, 708
Antenna control, 94
  cascade compensation and, 650–651
  closed-loop response design for, 272–276
  controller/observer design and, 704–709
  digital cascade compensator design and, 767–769
  gain design and, 650–651
  lag-lead compensation and, 508–511
  open-loop response and, 207–208
  root locus for, 426
  stability design/transient performance and, 606–607
  stability design via gain, 323–325
  state-space representation and, 144
  steady-state errors and, 368–371
  transient design via gain and, 426–427, 765–767
Aquifer system model, 147
Aquifers, 147–148
Aragula, S., 223
Arc welding robot, 439
Armature, 80
Armature circuit, 16, 80

Armature resistance, 81
Armature voltage, 80
  torque-speed curve with, 82
Artificial heart, open-loop transfer function, 223
Ashkenas, I., 233
Assembly-line robots, steady-state errors and, 354
Åström, K., 299, 452, 616, 623, 759, 780
Asymptotes
  break frequency, 542
  high-frequency, 542
  low-frequency, 542
  root locus sketching with, 400–402
Asymptotic approximations, 540–559
Atencio, A., Jr., 386
Atracurium, 221
Automated guided carts, for Chevrolet Volt electric vehicles, 657
Automatic generation control (AGC), 524
Automobile guidance system, 380
Ayers, J., 25, 31

Back electromotive force, 80
Backlash, 74, 88
Backlash effect, on load angular displacement response, 199
Bagchi, S., 337, 623
Bahill, A. T., 31
Bailey, F. N., 289, 299
Ball-drop experiment, 155
Ballard, R. D., 289, 336
Bandwidth, 581
Baratta, R. V., 234
Barkana, I., 377, 386, 661
Barlow, J. B., 532, 721
Barrows, B., 299
Bathtub water level, 25
Bechhoefer, J., 27, 31
Behavior at infinity, root locus sketching and, 399–402
Bell Telephone Laboratories, 5
Belvin, W. K., 452
Ben-Dov, D., 290, 299
Bennett, S., 4n, 31
Benningfield, L. M., 116
Bessemer, Henry, 5
Bicycle dynamics, 295
Bilinear transformations, in digital control systems, 746–747
Biological system, 95–97
Bittanti, S., 525, 531, 616, 623, 775, 780
Bittar, A., 333, 337
Black Hawk (UH-60A) helicopter, 382
Block diagram(s), 17–18
  of multiple subsystems, 236–237
Block diagram reduction, 18
  digital control systems and, 739–742
  by moving blocks, 243–245
  of sampled-data systems, 740–741
  via familiar forms, 242–243
Block diagram window, in LabVIEW program, 862–868
Block parameters, in MATLAB Simulink program, 843–845
Blood vessel blockages, 108
Bode, H.W., 5
Bode plots
  of (s+a), 542
  for $G(s) = (s+a)$, 541–544
  for $G(s) = 1/s$, 545–546
  for $G(s) = 1/(s^2 + 2\zeta\omega_\eta s + \omega^2\eta)$, 552–556
  for gain adjustment, 628
  gain margin/phase margin from, 579
  for $G(s) = s$, 545
  for $G(s) = s^2 + 2\zeta\omega_\eta s + \omega^2\eta$, 549–550
  for lead compensation, 639
  range of gain for stability via, 577–578
  for ratio of first-order factors, 546–549, 556–559
  for ratio of second-order factors, 550–552, 556–559
  static error constants from, 595–596
  transfer function from, 603–605
Bona, B. E., 160, 233, 300, 337, 722
Bondia, J., 160, 722
Borovic, B., 221, 232
Bottom-up design, 22
Bounded input, 302
Bounded-input bounded-output (BIBO), 302–303
Boutayeb, A., 115
Boyd, M., 780
Branches, root locus sketching and, 397
Breakaway points
  explanation of, 402–404, 412
  using differential calculus to find, 404
  without differentiation, 405–406
Break frequency, 542
Break-in points
  explanation of, 402–404, 412
  using differential calculus to find, 404
  without differentiation, 405–406
Bruner, A. M., 440, 452
Budak, A., 531
Butterworth polynomial, 331

Cai, Y., 386, 453, 623
Campbell, T. J., 452
Cancellation, pole-zero, 195
Cannon, R. H., Jr., 31, 115
Canonical form
  controller, 260–262
  Jordan, 260
  observer, 262–263
Capacitors, 130
Carlson, L. E., 115, 159
Cascade compensation
  antenna control and, 650–651

in digital control systems, 759
steady-state error design via, 626
steady-state errors via, 459–469
transient response improvement via, 469–481
Cascade compensation via *s*-plane, digital control systems and, 758–762
Cascade compensators, 494
Cascaded interconnections, 34
Cascaded subsystems, 238
Cascaded systems, load in, 238
Cascade form
of multiple subsystems, 237–239
of state space, 257–259
Cascade PI controller, robotic manipulator with, 447
Catenary, pantograph and, 110
Cavallo, A., 153, 159, 289, 299, 452
Cereijo, M. R., 159
Characterizing response, from damping ratio, 176
Charge-coupled device, in digital camera, 617–618
Chassaing, R., 763n, 780
Chebyshev, P.L., 5
Chemical process, temperature control system, 525
Chesmond, C. J., 380, 386
Chetouani, A., 115
Chevrolet Volt electric vehicles, automated guided carts for, 657
Chignola, R., 109, 115
Chiu, D. K., 153, 159
Cho, D., 153, 452, 531, 715, 721
Circuit, transformed, 49
Classical technique, 118. *See also* Frequency-domain technique
Cleveland, J. P., 233
Clifford, William Kingdon, 5
Closed-loop frequency responses, closed-loop transient responses and, 580–583
Closed-loop/open-loop frequency responses, relation between, 583–588
Closed-loop polar plot, 587n
Closed-loop poles, 303–305, 388–389, 392, 415, 418, 424, 461
Closed-loop response design, for antenna control, 272–276
Closed-loop step response, with LabVIEW program, 869–871
Closed-loop system, 9, 162
block diagram of, 8
error, 342
for ideal integral compensator, 461
root sensitivity of, 425
Closed-loop transfer function, sensitivity of, 362–363
Closed-loop transient responses
closed-loop frequency responses and, 580–583
open-loop frequency responses and, 589–592
Closed-loop vehicle response, for train stopping, 441
Cochin, I., 115, 159
Cochran, J. E., 332, 337, 661
Cockburn, J. C., 288, 299
Coefficient of viscous friction, 62, 70
Combustor, with microphone/loudspeaker, 445
Companion matrices, 262
Compensated system, root locus for, 462
Compensating zero, via rate feedback, 497–500
Compensation
active-circuit realization of, 504
physical realization of, 503–508
techniques, 458
Compensator, 9
passive realization of, 506
root locus with, 460
root locus without, 460
Completely controllable, 674n
Completely observable, 691n
Complex circuits
via mesh analysis, 51–52
via nodal analysis, 54–56
Complex numbers, vector representation of, 389–391
Complex pole, angle of departure/arrival from, 409
Component design, transient response through, 185–186
Component responses, of three-pole system, 188
Computer-aided design, 20–21
Computer-controlled systems, 9–10
Computer hard disk drive, 10
Computer simulation, of step responses, 429
Conductance, 54n
Conservation, flow for, 148
Constant-acceleration inputs, 341
Constant command, 18
Constant *M* circles, 583–584, 586
Constant *N* circles, 583, 585, 586
Constant-velocity inputs, 341
Continuous casting, in steel production, 294
Continuous stirred tank reactor, 715
Contours, 560–563
Control design palette, in LabVIEW program, 883–885
Controllability, 672–676
by inspection, 673–674
via controllability matrix, 675–676
Controllability matrix, 674
controllability via, 675–676
Controlled variable, 34
Controller canonical form, of state space, 260–262
Controller design, 665–672
alternative approaches to, 676–682
by matching coefficients, 677
for phase-variable form, 669–671
by transformation, 679–681
Controller/observer design, antenna control and, 704–709
Control system problem, for root locus, 388–389

Control systems. *See also* Feedback control systems; *specific control systems*
advantages, 3–4
analysis of, 10–15, 18–19
components, 2
configurations of, 7–10
derivative, 459
description of, 2
design, 18–19
digital computers in, 6
engineering, 21–22
history of, 4–7
integral, 459
prevalence of, 23
proportional, 459
schematic for, 16–17
test waveforms, 19
theory, 5
twentieth-century developments, 5
Control systems analysis, with LabVIEW program, 858–859
Control System Toolbox, 157, 228, 248, 297, 335, 384, 450–451, 530–531, 621, 660, 778–779
Convolution integral, 203
Cook, P. A., 115
Cooper, P. A., 386
Coordinate measuring machine, 525–526
Coulomb friction, 65n
Craig, I. K., 31, 110, 115, 156, 159, 228, 232, 299, 337, 383, 386, 452, 529, 531, 621, 623, 659, 662, 718, 721, 777, 780
Cramer's rule, 53
Crane loading, 107
Crawshaw, L. I., 32
Critically damped response, 171–172
Critical points, root locus sketching and, 412–414
Crosslapper, 224
Cubitt, William, 5
Cutaneous rabbit, 28
Cutting force control system, 332

D 'Alembert's principle, 17
D'Ambrosia, R. D., 234
Damped frequency of oscillation, 182
Damper, mass and, 125
Damping, 81
Damping frequency, exponential, 182
Damping ratio
characterizing response from, 176
definition of, 174
natural frequency and, 208
v. percent overshoot, 180
from phase margin, 589–591
v. rise time, 182
of second-order system, 173–175
second-order underdamped responses for, 178
Dancer, 25
Davis, S. A., 115
D'Azzo, J. J., 31, 337, 386, 531, 662, 721
Dc-dc converter, 225
Dc motor, 79
driving rotational mechanical load, 81
load and, 83–84
Dc servomotor, load and, 152
Deadzone effect, on load angular displacement response, 198
Decay frequency, exponential, 174
de Araújo, F. M., 116, 453, 623
Decoupled equations, 292
Decoupled variables, 125
Dee, A., 288, 299
Deflection response, of fluid-filled catheter, 222
Degrees of freedom, 65
in rotational systems, 70
De Keyser, R., 299
Dell'Orto, F., 531, 623, 780
De Maria, G., 153, 159, 299, 452
Dependent source, electrical network with, 128–130
Derivative compensation, ideal, 470–477
Derivative control systems, 459
Design
computer-aided, 20–21
with LabVIEW program, 858–859
objectives, 10–15
theory, 5
Design process
flowchart of, 15
steps of, 15–20
Design via frequency response, with MATLAB program, 818–823
Design via root locus, with MATLAB, 808–813
Design via state space, with MATLAB program, 823–826
Desoer, C. A., 337
de Vlugt, E., 292, 299
Destination shaft, 76
Diabetes population model, 105
Diagonalizing
system in state space, 270–271
system matrix, 268
DiBona, G. F., 222, 232
Di Carlo, A., 531, 623, 780
Differential calculus, 404
Diesel power distribution system, hybrid solar cell and, 294
Differential equation
coefficients, 5
Laplace transform solution of, 39–40
linearization of, 91–92
single loop via, 48–49
transfer function for, 45
Digital camera, charge-coupled device in, 617–618
Digital cascade compensator design, 760–761
antenna control and, 767–769

Digital compensator, implementing, 762–765
Digital computers
  advantages, 725–726
  in control systems, 6
  modeling, 727–730
  placement within loop, 725
Digital control systems, 723–725
  bilinear transformations in, 746–747
  block diagram reduction and, 739–742
  cascade compensation in, 759
  cascade compensation via *s*-plane and, 758–762
  digital system stability in, 742–749
  gain design on *z*-plane and, 755–758
  with MATLAB program, 827–833
  with MATLAB Simulink program, 852–855
  stability design via root locus in, 755–756
  steady-state errors and, 749–753
  transfer functions of, 735–739
  transient response design via gain adjustment and, 756–757
  transient response on *z*-plane, 753–755
  z-transform and, 730–735
Digital feedback control system
  steady-state errors for, 750
  unit parabolic input for, 751
  unit ramp input for, 751
  unit step input for, 750–751
Digital numerical control, lathe with, 742
Digital system simulation, with LabVIEW program, 881–885
Digital system stability
  digital control systems in, 742–749
  via Routh-Hurwitz, 748
  via *s*-plane, 747–748
  via *z*-plane, 742–743
Digital-to-analog conversion, 729
Digital versatile disc players, 525
Dirac delta functions, 729
Disk drive arm, 447
Disk drive position control system, 442
Distributed parameter, 71n
Disturbances, steady-state errors for, 356–358
Doebelin, E. O., 31, 115
Dominant-pole argument, 223
Dorf, R. C., 31, 115, 193, 232, 337, 453, 531, 623, 662
Drebbel, Cornelis, 4
Driving simulator, 642
DVD player, 7
Drug absorption, 146–148
D'Souza, A. F., 26, 31, 115
Duals, 263
Dynamic voltage restorer (DVR), 378, 441
Dynamometer, 26
Dynamometer test, 81, 82

Economics, as design consideration, 12
ECU. *See* Electronic control unit
Edelstein-Keshet, L., 108, 109, 115
Eigenvalues, 268
  poles and, 200–202
  transfer function poles and, 200–202
Eigenvector, 268, 269
Ekeberg, Ö., 159
Electrical constants, of motor transfer function, 82
Electrical network
  branch currents in, 126
  with dependent source, 128–130
  representation of, 126–127
  transfer function, 46–61
Electrical to mechanical systems analogies, 62
Electric circuit analogs, 84–87
Electric network, 28
Electric ventricular assist device (EVAD), 292, 776
Electromagnetic systems, 79–84
Electromechanical system transfer functions, 75–80
Electronic control unit (ECU), 30
Elevator response, 2
Elevators, 3
El-Gamal, M., 337, 623
Elkins, J. A., 115, 159
El-Samad, H., 453
Emami-Naeini, A., 31, 115, 159, 232, 453, 623, 721
Energy storage elements, 126
Enzyme breakdown, 107
Eppinger, S. D., 32, 116, 160, 233, 300, 337, 386, 453, 532, 624, 662, 721, 781
Epsilon method, stability vi, 308–309
Equations of motion, 63, 71
  by inspection, 67–68, 72–73
Equilibrium, 89
Errors, 9. *See also* Steady-state errors
Estimator, 682
EVAD. *See* Electric ventricular assist device
Evans, J., 288, 299
Evans, Walter, 5
Existing transient response, 10
Exponential damping frequency, 182
Exponential decay frequency, 174
Exponential frequency, 167
Exponential response, 164
Exponential time constant, 174

F4-E military aircraft
  pitch stabilization loop, 444
  steering control, 153
Factoring, via Routh-Hurwitz, 319
Fagergren, A., 155, 159
Fahlén, P., 337, 624
FANUC M-410iB robot, 323
Feedback amplifier analysis, 5

Feedback compensation, 495–503
approach 1, 496–500
approach 2, 500–503
generic control system, 495
minor-loop, 500–503
Feedback control systems, 9, 240. *See also* Control systems; *specific control systems*
analysis/design of, 245–246
for ideal derivative compensation, 472
with MATLAB Simulink program, 850–852
Feedback form, of multiple subsystems, 240–241
Feedback path, 9
Feedback system simulation, with LabVIEW program, 880–881
Fenn's law, 291
Fertilizer applicator, 448
Final value theorem, steady-state errors using, 365
Finances, of control systems, 12
First-order system, 166–168
poles of, 163
unit step and, 166
zeros of, 163
First-order transfer functions via testing, 167
Fission reaction, 25
Fixed field, 80
Floor vibration eliminator, 447
Floppy disk drive, block diagram, 617
Flower, T. L., 662
Flow for conservation, 148
Fluid-filled catheter, deflection response, 222
Flyball speed governor, 5
Force-displacement, 62
Forced response, 11, 162, 164
Force-velocity, 62
Foroni, R. I., 115
Forssberg, H., 159
Forward-path gain, 251
Forward-path transfer function, 445
Forward transfer function, 339. *See also* Open-loop transfer function
Frankle, J. T., 32
Franklin, G. F., 31, 115, 159, 231, 453, 623, 721
Free-body diagram, 63
transformed, 63
Free viruses, 110
Frequency, exponential, 167
Frequency domain modeling, with MATLAB program, 788–793
Frequency-domain technique, 118. *See also* Classical technique
Frequency/gain, at imaginary-axis crossing, 406–407
Frequency response
analytical expressions for, 536–537
concept of, 535–536
lead compensator, 636–637
with MATLAB program, 813–818
sinusoidal, 535
steady-state errors from, 593–597
techniques, 533–608
from transfer function, 538–539
Frequency response design methods
antenna control and, 650–652
lag compensation and, 630–635
lag-lead compensation and, 641–650
lead compensation and, 635–641
overview of, 627
root locus and, 626
transient response via gain adjustment and, 627–630
Frequency response plots, 537–538
of time delay systems, 598–599
Friction, 65n
Fuel-cell power plants, 6
Functional block diagram, 14
drawing of, 16
Functions, linearizing, 90

$G(s)$, steady state error as, 345–348
Gain-adjusted antenna control system, step responses of, 427
Gain adjustment
bode plots for, 628
stability/transient response design via, 626
transient response design via, 415–419, 626
transient response via, 627–630
Gain design
antenna control and, 650–651
for transient response, 247–248
Gain design on z-plane, digital control systems and, 755–758
Gain margin/phase margin
from Bode plots, 579
evaluating, 578
via Nyquist diagram, 574–576
Galvæo, R.K.H., 115, 453, 616, 623
Gamble, J. B., 26, 32, 300
Gas-fired heater, block diagram, 717
Gauthier, M., 234
Gear backlash, 88
Gear driven rotational systems, 75
Gear systems, 74–78
with loss, 77–78
motion equation for, 77
schematic, 74
Gear train, 77
Geselowitz, D. B., 300, 662, 781
Ghonien, A. F., 452
Gillard, D., 234
Glantz, A. S., 222, 232
Global carbon cycle, schematic description of, 155
Golbon, N., 532, 781
Gompertz growth model, 109
Gong, W., 386, 453, 623, 662

Good, M. C., 233, 623
Graham, D., 233
Graphical user interface (GUI), 196
Griggs, G. E., 115, 159
Graphical user interface (GUI), 20, 21
Grinder system, 26
GUI. *See* Graphical user interface
Guy, W., 453
Gyro systems, 151

Hacisalihzade, S. S., 32
Hammel, H. T., 32
Han, Y. D., 386, 453
Harbor, R. D., 337, 662
Hardy, H. L., 439, 453, 774, 780
Harmonic drives, robotic manipulator with, 448
Hatopoglu, C., 454
Head, 147
Heat-exchanger process, 521–522
Hedrick, J. K., 452
Hekman, K. A., 333, 337
Heller, H. C., 32
HelpMate transport robot, 288–289
Hess, R. A., 382, 386
High-frequency asymptotes, 542
High-speed proportional solenoid valve, 26
High-speed rail pantograph control problem, 28–29, 109–110, 155–156, 227, 296–297, 334, 383, 449, 528, 620–621, 659, 718, 777
HIV/AIDS control system problem, 29, 110–111, 156, 227–228, 297, 334–335, 383, 449–450, 528–529, 621, 659, 718–719, 777
Hogan, B. J., 32
Hold
  zero-order, 728
  zero-order sample-and-, 726, 730, 737–738
Hollot, C. V., 386, 441, 453, 623, 662
Home entertainment systems, 6
Homogeneity, 88
Hong, J., 152, 160
Horizontal canards, 153
Hormonal regulation mechanisms, 441
Horta, L. G., 452
Hospital pharmacy robot, 664
Hostetter, G. H., 32, 299, 337, 386, 531, 623, 662, 720, 780
Houpis, C. H., 31, 337, 386, 531, 662, 720
Hsu, J. C., 115
Hubble Space Telescope (HST), 619
Human eye movement, 288
Human growth population, 108
Human leg, transfer function of, 95–96
Human postural dynamics, 107
Human response, to visual cue, 220
Hybrid solar cell, diesel power distribution system and, 294
Hybrid vehicle control problem, 29, 111–112, 156–157, 228, 297–298, 335, 384, 529, 621, 660, 719, 777–778
Hydraulic actuator, 26
Hydraulic circuits, 227
Hydraulic pumps, 227
  diagram of, 227
Hypothalamic-pituitary-adrenal axis, linear time-invariant model of, 154

ICE. *See* Internal combustion engine
Ideal compensators, 458
Ideal derivative compensation, 469, 470–480
  design, 472–476
  feedback control system for, 472
  predicted characteristics for, 471
  root locus for, 473, 475
  uncompensated/compensated system, 472–473
Ideal integral compensated system response, uncompensated system response and, 463
Ideal integral compensator, 459, 461
  closed-loop system for, 461
  effect of, 461
  implementing, 463
Ideal sampling, zero-order hold and, 730
Identity matrix, 139
Ignatoski, M. A., 300, 622, 781
Imaginary-axis crossing, frequency/gain at, 406–407
Impedance for mechanical components, 64
Impedance relationships, 48
Impedance rotational relationships, 70
Impedance translational relationships, 62
Implantable medical devices, with in-body batteries, 447
Impulse, 18, 19
In-body batteries, implantable medical devices with, 447
Inductors, 130
Industrial robots, 200–221, 617. *See also* Robots
Inertia, 17, 70, 81
  moment of, 70
Inherent disturbance suppression, 775
Inigo, R. M., 160
Initial conditions, 45
Inner ear hair cell transducers, 105
Innocenti, M., 337, 661
Input, 2
Input potentiometer, 94
Input signals
  root locus and, 421
  test of, 18
Input substitution
  analysis via, 366–367
  steady-state errors using, 367
Input transducer, 8
Insect flight dynamics, 225
Inspection, mesh equations via, 56–57
Instability, 11, 302, 303–304. *See also* Stability

Insulin concentration-time evolution, 153–154
Integral control
  design, 701–703
  steady-state error design via, 700–704
  systems, 459
Intelligent Soft Arm Control (ISAC), 331, 332
Interconnection of subsystems, block diagram of, 34
Internal combustion engine (ICE), 29
Inverse Laplace transform, 35, 36, 166
Inverse z-transform
  via partial-fraction expansion, 733–734
  via power series method, 734
Inverting operational amplifier, 58
  circuit, 59
  schematic, 58
Ionescu, C., 233
Ionic polymer-metal composite, 225
ISAC. *See* Intelligent Soft Arm Control
Isailovic, J., 386
Isoflurane, 221

Jannett, T. C., 221, 233
Jason, 317
Jason Junior, 272
Jenkins, H. E., 26, 32
Johansson, R., 116
Johnson, H., 233, 289, 299, 337, 453, 531, 780, 800, 835
Johnson, R. E., 227, 233
Joint flexibility, model representing, 448
Jordan canonical form, 260
Juang, J., 452
Jury's stability test, 745
$j\omega$-axis crossings, root locus sketching and, 405–407

Kailath, T., 35, 116, 160, 721
Kandel, A., 160
Kanellakopoulos, I., 107, 116, 291, 300, 386
Kara, A., 337, 616, 623
Kato, Y., 531, 721
Katz, P., 781
Kawamura, K., 332, 337, 623
Kermurjian, A., 116
Kesson, M., 116
Khaminash, M., 453
Kharitonov's theorem, 330
Khodabakhshian, A., 524, 532, 776
Kirchhoff's current law, 17, 47, 84
Kirchhoff's nodal equations. *See* Nodal equations
Kirchhoff's voltage law, 17
Klaasens, J. B., 234, 722
Klapper, J., 32
Klein, R. E., 299, 452, 623
Knight, B., 234
Koontz, J. W., 300, 662, 781
Kostis, J. B., 109, 116
Krishnamurthy, B., 299
Ktesibios' water clock, 4, 290
Kumar, R. R., 380, 386
Kuo, B. C., 233, 453, 532, 623, 662, 745n, 781
Kuo, F. F., 116, 623
Kurfess, T. R., 26, 32
Kurland, M., 287, 299
Kwatny, H. G., 453, 662
Kyrylov, V., 154, 160, 714, 721

LabVIEW program, 21
  amplifier saturation simulation with, 878–880
  analysis examples with, 862–875
  block diagram window in, 862–868
  closed-loop step response with, 869–871
  control design palette in, 883–885
  control systems analysis with, 858–859
  design examples with, 862–875
  design with, 858–859
  digital system simulation with, 881–885
  feedback system simulation with, 880–881
  linear systems simulation with, 876–878
  open-loop/closed-loop sinusoidal frequency analysis/design with, 873–875
  open-loop step response with, 862–869
  root locus analysis/design with, 871–872
  simulation palette in, 881–882
  simulation with, 858–859
  using, 859–862
Lag compensation, 464–469, 630–635
  design procedure, 631–634
  root locus and, 465
  type 1 systems, 464
  visualizing, 630–631
Lag compensator, 459
  design, 466–468
Lag-lead compensation, 641–643
  antenna control and, 508–511
  design procedure, 643–646
  using Nichols chart, 646–649
Lag-lead compensator, 487
  design, 487–491
  ramp response error for, 491
  root locus for, 488, 489, 490
Lago, G., 116
Lam, C. S., 379, 386, 453
Landesberg, A., 300
Laplace transform, 17, 35–36
  definition of, 35
  of differential equation, 39–40
  inverse, 35, 36
  review, 35–44
  of state-transition matrix, 203–204
  state-transition matrix via, 206–207
  table, 36

theorems, 37
of time function, 35–36
Laplace transformed circuit, 49
Laplace transform solution, 200–202
of state equations, 198–202
Lathe, with digital numerical control, 742
Lead compensation, 477–481
Bode plots for, 639
design, 478–481, 637–640
geometry of, 477
possible solutions, 478
visualizing, 635–636
Lead compensator, 469
frequency response, 636–637
realization, 507
Ledgerwood, B. K., 115
Lee, Edmund, 4
Lee, S., 153, 159
Lennartsson, A., 299, 452, 623
Leo, D. J., 233
Lepschy, A. M., 289, 300, 377, 386
Lewis, F. L., 222, 232
Liang, S. Y., 337
Liceage-Castro, E., 154, 160
Liew, K. M., 386, 453, 623
Lim, T. W., 386
Lin, J.-S., 107, 116, 291, 300, 379, 386
Lin, R., 338, 454, 662
Linear combination, 121
Linear control systems analysis, 5
Linear independence
explanation of, 123
state variables and, 123–124
Linear systems, 88
Linear system simulation
with LabVIEW program, 876–878
with MATLAB Simulink program, 842–848
Linear time-invariant differential equation, 17
Linear time-invariant model, of hypothalamic-pituitary-adrenal axis, 154
Linear voltage differential transformer (LVDT), 26, 223
Linearity, 88
Linearization, 89–97
about a point, 89
of differential equations, 91–92
of function, 90
state-space representation and, 141–143
Linearized magnetic levitation system block, 440
Linearly dependent motions, 65
Linearly increasing command, 19
Linearly independent, 122
state variables, 124–125
Linkens, D. A., 221, 233
Liquid-level control, 4
Lithium-ion battery charger, 333
Liu, A. Q., 222, 232
Load
in cascaded systems, 238
dc motor and, 83–84
dc servomotor and, 152
motor and, 95, 144–145
tester, 289
Load angular displacement response
backlash effect on, 199
deadzone effect on, 198
Load angular velocity response, amplifier saturation and, 197
Log-magnitude plots, 541–542, 551–552, 576–577
Longitudinal flight model linearization, 224
Look-ahead offset, 334
Loop analysis. *See* Mesh analysis
Loop gain, 241, 251
Loops
major, 496
minor, 495, 496
nontouching, 251
Lordi, N. G., 160
Lossless gears, 75
angular displacement in, 75
system with, 76–77
Low back pain, motor trunk patterns and, 223
Low, K. H., 386, 453, 616, 623
Low-frequency asymptotes, 550
LTI Viewer, 217, 229, 230, 575, 579
Ludwick, S. J., 26, 32
Luenberger, D. G., 721
Lumelsky, V., 299
Lumped parameter, 71n
Lusitania, 317
Lyapunov, Alexandr Michailovich, 5

Mablekos, V. E., 116
Magnetic levitation transportation system, 332, 526–527
Magnitude frequency response, 536
Magnitude plots, 537
Magnusson, M., 116
Mallavarapu, K., 225, 233
Malsbury, T., 386
Malthus, Thomas, 108
Manned submersible, 272
Manring, N. D., 227, 233
Mapping contour, 560–561
Mapping only positive $j\omega$-axis, stability via, 571–573
Marginally stable systems, 303, 304
Marginal stability, 302
Martin, R. H., Jr., 32
Marttinen, A., 108, 116, 330, 337, 616, 623
Mason, S. J., 251, 300
Mason's rule
multiple subsystems and, 251–254
transfer function via, 252–253

Mass, 62, 64
  damper and, 125
Matching coefficients, controller design by, 677
Mathematical models, 17–18
  from physical system schematics, 34
MathWorks Inc., 20
MATLAB program, 20–21
  command summary for, 833–835
  design via frequency response with, 818–823
  design via root locus with, 808–813
  design via state space with, 823–826
  digital control systems with, 827–833
  examples, 788–833
  frequency domain modeling with, 788–793
  frequency response techniques with, 813–818
  M-files with, 787
  multiple subsystems with, 799–804
  root locus with, 806–808
  stability with, 804–805
  time domain modeling with, 793–795
  time response with, 795–799
  using, 787
MATLAB Simulink program
  accessing, 837
  block parameters in, 843–845
  creating new model with, 838
  digital control systems with, 852–855
  examples, 841–855
  feedback systems with, 850–852
  libraries, 838–841
  linear system simulation with, 842–848
  saturation nonlinearity with, 848–850
  simulation parameters in, 846–847
  using, 836–841
Matrices
  companion, 262
  controllability, 674–676
  observability, 691–693
  state-transition, 203, 204, 206–207
  system, 268, 270–271
  transformation, 266, 268, 270, 272
Maxwell, James Clerk, 5
May, F. P., 722
Mayr, O., 4n, 32
McRuer, D., 224, 233
Mechanical constants, 81
Mechanical displacement, 62
Mechanical system
  into parallel analog, 87
  into series analog, 86
Mechanical system transfer functions
  rotational, 69–74
  translational, 61–69
Meier, R., 32
MEMS (Micro Electromechanical System), 221
Mendoza, C., 116
Mesh, 48–49
Mesh analysis
  complex circuits via, 51–52
  single loop via, 48–50
Mesh equations, via inspection, 56–57
Meyer, A. U., 116
M-files, with MATLAB program, 787
Mian, G. A., 290, 300, 386
Michaelis-Menten equations, 107
Micro Optical Electromechanical Systems (MOEMS), 221
Milhorn, H. T., Jr., 288, 300, 721
Milsum, J. H., 105, 116
Minnichelli, R. J., 300, 337
Minor loop, 495, 496
Minor-loop feedback compensation, 500–503
  root locus for, 502
  step response simulation for, 503
Minorsky, Nicholas, 5
Misra, V., 386, 453, 623, 662
Missile control system
  modeling of, 743–745
  stability of, 743–745
Missile steering control, 152
MOEMS. *See* Micro Optical Electromechanical Systems
Monahemi, M. M., 532, 721
Moment of inertia, 70
Momentum wheel, pitch axis attitude control system with, 444
Motion equation, for gear systems, 77
Motors
  explanation of, 79–80
  transfer functions and, 95–96
Motor, load and, 95, 144–145
Motorcycle radio volume, 25
Motor dead zone, 88
Motor transfer function, electrical constants of, 82
Motor trunk patterns, low back pain and, 223
Mott, C., 27, 32
Multiple loops, 52–53
Multiple nodes, 54
  with current sources, 55–56
Multiple root of multiplicity, 40
Multiple subsystems. *See also* Subsystems
  background on, 236
  block diagrams of, 236–237
  cascade form of, 237–239
  feedback form of, 240–241
  Mason's rule, 251–254
  with MATLAB program, 799–804
  parallel form of, 239
  signal-flow graphs of, 248–251
  similarity transformations, 266–277
  in space shuttle, 237

Multiplicity, 40
Murakami, T., 386
Muscle relaxation, 221

Nafion sheet, 225
Nagle, H. T., 233, 781
Nakamura, M., 223, 233
Nano-positioning device, 223
NASA flight simulator robot arm, 79
Nashner, L. M., 221, 233
National Instruments PXI, 534
Natural frequency
  damping ratio and, 208
  of general second-order system, 172
Natural period, 174
Natural response, 11, 162, 164
n-channel enhancement-mode MOSFET Source Follower circuit, 280–281
Neamen, D. A., 291, 300
Negative feedback, 240n
Negative-feedback systems, 421
Negative step response, of pitch control, 212
Network theory, 127
Newbury, K., 233
Newton's laws, 17, 63
Nichols charts, 587–588
  lag-lead compensation design using, 646–649
Nieuwland, J., 32
Nilsson, J. W., 116, 624
No, T. S., 337, 661
No-load speed, 82
Nodal analysis
  complex circuits via, 54–56
  simple circuits via, 50–51
Nodal equations
  form of, 56
  method to write, 54, 55
Nodes, of signal-flow graphs, 248–249
No integration systems, steady state error for, 347–348
Noload speed, 82
Noninverting operational amplifier, 59–60
  circuit, 60
  schematic, 60
Nonlinear electrical network, 92–93
Nonlinearities, 88–89
  physical, 88
  time response and, 196–198
Nonlinear systems, 88
  representation of, 142–143
Nonlinear translational mechanical system, 143
Nonminimum-phase electric circuit, 193
Nonminimum-phase system
  step responses of, 192, 194
  transfer function of, 192–194
Nontouching loops, 251–252
Nonunity feedback systems
  steady-state actuating signal for, 361
  steady-state errors for, 358–362
Norton's theorem, 55
Notch filter, 492–494
  root locus for, 492–493
Novosad, J. P., 32
*n*th-order differential equations
  converted to *n* simultaneous first-order differential equations, 120
  explanation of, 17–18
Nwokah, O.D.I., 321
Nuclear power generating plant, 25
Nuclear reactor, 25
Nyquist, H, 5
Nyquist criterion, 559–576
  derivation of, 560–563
  range of gain for stability via, 570–571
  stability determination with, 563–564
Nyquist diagram
  gain margin/phase margin via, 574–576
  for open-loop function, 567–569
  sketching, 564–569
  stability via, 569–573
Nyquist sampling rate, 726

Observability, 689–693
  by inspection, 690
  via observability matrix, 691–692
Observability matrix, 691
  observability via, 691–692
  unobservability via, 692–693
Observer, 682
Observer canonical form
  observer design for, 686–688
  of state space, 262–265
Observer design, 682–689
  alternative approaches, 693–700
  by matching coefficients, 697–699
  for observer canonical form, 686–688
  by transformation, 695–697
O'Connor, D. N., 28, 32, 109, 116, 160, 227, 233, 296, 300, 334, 337, 383, 386, 449, 453, 528, 532, 620, 624, 659, 662, 718, 777, 781
Octave, 541
Ogata, K., 32, 116, 233, 532, 624, 662, 674n, 678n, 691n, 721, 726n, 750n, 759n, 781
Ohnishi, K., 383, 386
O'Leary, D. P., 532, 721
OMS. *See* Orbital maneuvering system
One integration systems, steady state error for, 348–349
Open-loop/closed-loop sinusoidal frequency analysis/design, with LabVIEW program, 873–875
Open-loop frequency responses
  closed-loop transient responses and, 589–592
  response speed from, 591–592

Open-loop function, Nyquist diagram for, 567–569
Open-loop pitch response, UFSS and, 207–208
Open-loop poles, 381–382, 385, 391, 399–400, 441
Open-loop response, antenna control and, 207–208
Open-loop step response, with LabVIEW program, 862–869
Open-loop systems, 8–9
  block diagram of, 8
Open-loop transfer function, 241
  for artificial heart, 223
Operational amplifier, 58, 193
  schematic of, 58
Optical disk recording system, 6
Orbital maneuvering system (OMS), 6
Oscillations
  damped, 14–15, 174
  damped frequency of, 170
O'Sullivan, D. M., 234, 722
Output, 2
Output engine fan speed, 330
Output equation, 121, 123
Output potentiometer, 94
Output response, 162
Output transducer, 9
Overdamped behavior, 213
Overdamped response, 170, 172
Overshoot, 178, 410, 416–417, 600–601
özel, T., 526, 532
özgüner, Ü., 337, 454

Packet information flow, 383
Padé approximation, 334, 382
Palazoglu, A., 532
Pantograph, catenary and, 110
Pantograph head displacement, 110
Papin, Denis, 4
Papson, T. P., 299
Parabolic inputs, 19
  steady state error and, 347
Parallel analog, 86–87
  development of, 87
  mechanical system converted into, 87
  parameters for, 87
Parallel form
  of multiple subsystems, 239
  of state space, 259–260
Parallel hybrid-electric vehicle, 29–30
Parallel subsystems, 238
Paramagnetic oxygen analyzer, 380
Partial-fraction expansion, 37–44
  inverse z-transform via, 733–734
Particular solutions, 11, 162n
Passive-circuit realization, 506–507
PD controllers. *See* Proportional-plus-derivative controllers
Peak time, 178
  evaluation of, 179
  lines of constant, 183
  from pole location, 184
  from transfer function, 181–182
Pendulum, simple, 142
Percent overshoot, 178
  v. damping ratio, 180
  evaluation of, 180
  lines of constant, 183
  from pole location, 184
  for time delay systems, 600–601
  from transfer function, 181–182
Performance, 2
Pérez López, O., 721
Pfleiderer, H. J., 160, 722
Pharmaceutical drug absorption, 146–147
Pharmaceutical drug-level concentrations, 146
Phase frequency response, 536
Phase margin, damping ratio from, 589–591
Phase shift oscillator, 333
Phase-variable representation
  controller design for, 669–671
  for plant, 667
Phase variables, 133
  block diagram of, 135
  choice, 133
Phasors, 535
Philco Technological Center, 160
Phillips, C. L., 233, 337, 662, 781
Philon of Byzantium, 4
pH processes, modeling/control of, 224
Phugoid mode, 224
Physical system, transform requirements into, 16
Physical system schematics, mathematical models from, 34
Pickoff points, 236
Picó, J., 160, 722
PI controllers. *See* Proportional-plus-integral controllers
PID controllers. *See* Proportional-plus-integral-plus-derivative controllers
Piecewise linear systems, 104
Pinette, B., 152, 160
Pioneer Electronics, Inc., 7
Piper, G. E., 444, 453, 662
Pitch angle control representation, in UFSS, 276–277
Pitch axis attitude control system, momentum wheel with, 444
Pitch control loop
  root locus of, 428
  for UFSS vehicle, 211
Pitch control loop without rate feedback, step responses, 429
Pitch control loop with rate feedback, step responses of, 430
Pitch control system
  negative step response of, 212
  for space shuttle, 287–288
Pitch gain, 324
Pitch stabilization loop, of F4-E military aircraft, 444
Plant

phase-variable representation for, 667
pole placement for, 668–669
state-space representation of, 666
with state-variable feedback, 706
Plate dispenser, 105
Polar plot, 587n
Pole(s), 162–163
eigenvalues and, 200–202
evaluating response with, 165
of first-order system, 163
imaginary part of, 214
real part of, 214
of transfer function, 163
underdamped response using, 171
Pole distribution, via Routh table with row of zeros, 312–313
Pole location
peak time from, 184
percent overshoot from, 184
for plant, 668–669
for root locus, 393
settling time from, 184
topology for, 666
Pole plot, for underdamped second-order system, 182
Pole sensitivity, root locus and, 424–425
Pole-zero plot, 164
Popa, D., 232
Position constant, 593–594
Position control system, 12, 34
antenna azimuth, 13–15, 208
response of, 14
tachometer as, 495
Positive-feedback systems, root locus for, 421–424
Postural arm reflexes, 292–293
Potentiometer, 9, 24
Powell, J. D., 31, 115, 159, 232, 453, 623, 721
Power amplifier, 94, 144
Power series method, inverse z-transform via, 734
Preamp, 94
Precision grip dynamics, 155
Prewarping, 759
Process control industry, 6
Processes, 2, 8
Prochazka, A., 234
Proportional control system, 459
Proportional-plus-derivative (PD) controllers, 469
Proportional-plus-integral (PI) controllers, 459, 464
Proportional-plus-integral-plus-derivative (PID) controllers, 5
characteristics of, 484
design, 482–486
implementing, 505
root locus for, 483
Pulse transfer function
derivation of, 736–737
of feedback system, 740–741
Pupil dilation, 26–27
Pythagorean theorem, 182

Qualitative analysis and design, 162
Qualitative method, 162
Quantization error, 726
Quarter-car model, for suspension design, 107

Radio antenna, 12
Raible's tabular method, 745
Ramp, 19
Ramp input
steady state error and, 346–347
steady-state error and, 341, 342
Ramp response error, for lag-lead compensator, 491
Random-access memory (RAM), 354
Random early detection (RED) algorithm, 383, 441, 656
Range of gain for stability
for time delay systems, 599–600
via Bode plots, 577–578
via Nyquist criterion, 570–571
Range of sampling interval, for stability, 745–746
Rate feedback, compensating zero via, 497–500
Raven, F. H., 116, 662
RCS. *See* Reaction control system
Reaction control system (RCS), 6
Read/write head displacement, 330
Real-axis breakaway, root locus sketching and, 402–405
Real-axis break-in points, root locus sketching and, 402–405
Real-axis pole, transient response and, 165
Real-axis segments, root locus sketching and, 398
Realization
active-circuit, 404–405
of lag-lead compensator, 509
passive-circuit, 506–507
Reciprocal, of time constant, 167
Reference, 8
Reference input, 34
Remote-controlled robot, 3
Residues, 195
Resistance, 49
Resistor, 47
Response, exponential, 164
Response speed, from open-loop frequency response, 591–592
Retinal light flux, 443
Reverse coefficients, stability via, 310
Richon, J.-B., 452
Riedel, S. A., 116
Riegelman, S., 160
Riseman, E. M., 152, 160
Rise time, 167, 178
v. damping ratio, 182
evaluation of, 181–183
from transfer function, 181–182
Rober, S. J., 332, 337

Robotic manipulator
with cascade PI controller, 447
harmonic drives with, 448
target environment and, 153
Robots, 3. *See also specific robots*
arc welding, 439
assembly-line, 354
FANUC M-410iB, 323
hospital pharmacy, 664
industrial, 220, 617
input commands to, 187
leg of, 142
remote-controlled, 3
with television imaging system, 152
transport, 288–289
vacuum, 220
walking, 141
Robust design, 12
Rockwell International, 6, 30, 691
Romagnoli, J. A., 532, 775, 781
Room temperature control, 521
Root locus, 5
for antenna control, 426
for compensated system, 462
with compensator, 460
without compensator, 460
control problem for, 388–389
definition, 388, 392–394
design via, 455–459
from general control system, 394
generalized, 419–421
for ideal derivative compensation, 473, 475
lag compensation and, 465
for lag-lead compensator, 488, 489, 490
with MATLAB program, 806–808
for minor-loop feedback compensation, 502
for notch filter, 492–493
for PID controllers, 483
of pitch control loop, 428
plotting/calibrating, 410
pole location for, 393
pole plot for, 393
pole sensitivity and, 424–425
for positive-feedback systems, 421–424
properties of, 394–397
sample, 456
for security camera system, 392–393
starting/ending points and, 398–399
for uncompensated system, 462
vector representation of complex numbers and, 389–391
Root locus analysis/design, with LabVIEW program, 871–872
Root locus sketching
angles of departure/arrival in, 407–409
with asymptotes, 400–402
behavior at infinity, 399–402
branches and, 397
critical points and, 412–414
$j\omega$-axis crossings and, 405–407
real-axis breakaway and, 402–405
real-axis break-in points and, 402–405
real-axis segments and, 398
refining, 402–411
rules for, 411–412
symmetry and, 397–398
Root sensitivity, of closed-loop system, 425
Roots of denominator of F(s)
complex or imaginary, 41–44
real and distinct, 38–40
real and repeated, 40–41
Rotational mechanical impedances, 76
Rotational mechanical load, DC motor driving, 81
Rotational mechanical system transfer functions, 69–74
Rotational systems
degrees of freedom in, 70
driven by gears, 75
Rotor, 80
Routh, Edward John, 5
Routh-Hurwitz criterion
digital system stability via, 748
examples of, 314–320
factoring via, 319
special cases of, 308–313
of stability, 305–308
with zero in first column, 314–315
Routh table
generating, 306–307
interpreting, 307
pole distribution via, 312–313
with row of zeros, 312–313, 316–317
stability via, 310–311
zero in, 308–311
Rover, 3
Rubbertuators, 331
Running resistances, 111

Safety valve, 4
Salapaka, M. V., 223, 233
Salapaka, S., 233
Salcudean, S. E., 290, 299
Sales, R. M., 337
Salminen, R. T., 108, 116, 337, 623
Sampled-data system. *See* Digital control systems
Sampled-data systems
block diagram reduction of, 740–741
transfer functions and, 735–739
z-transform and, 739
Sampler, modeling, 728–729
Sarcomere, 292
Saturation nonlinearity, with MATLAB Simulink program, 848–850

Savant, C. J., Jr., 32, 299, 337, 386, 531, 623, 662, 721
Savaresi, S. M., 531, 623, 780
Sawusch, M. R., 233
Scanning probe microscope, 775
Schematic, for control system, 16–17
Schierman, J. D., 330, 337
Schmidt, D. K., 337
Schneider, R. T., 220, 233, 383, 386
Schnell, S., 116
Schouten, A. C., 292, 299
Sebastian, A., 233
Second-order step response approximation, 195
Second-order step response components, 170
Second-order system damping cases, step responses for, 172
Second-order systems, 169–177
  damping ratio of, 173–175
  natural frequency of, 172
Second-order transfer functions, via testing, 186
Second-order underdamped responses
  for damping ratio values, 178
  specifications, 178
Second-order underdamped systems, step responses of, 184
Security camera system, root locus for, 392–393
Seering, W. P., 28, 32, 110, 116, 160, 233, 300, 337, 386, 453, 532, 624, 662, 721, 781
Segway personal transporter, 27
Self-guiding vehicle, bearing angle control of, 527
Semmlow, J. L., 109, 116
Sensitivity
  of closed-loop transfer function, 362–363
  steady-state errors and, 362–364
Sensor, 9
Serial hybrid-electric vehicle, 29
Series analog, 85–86
  mechanical system converted into, 86
Series RLC electrical network, 49
Settling time, 167, 178
  lines of constant, 183
  from pole location, 184
  from transfer function, 181–182
Severyanova, L. A., 160, 721
Shaw, D. A., 32
Shibata, M., 386
Shin, Y. C., 337
Shinners, S. M., 721
Ship roll axis, 212
Ship stability, 5
Ship steering, 5
Shortening muscle velocity, 292
Short period mode, 224
Signal-flow graphs
  components, 248
  converting block diagrams to, 249–250
  development stages of, 254
  of multiple subsystems, 248–251
  of state equations, 254–256
Similarity transformations
  of multiple subsystems, 266–277
  on state equations, 267
Simple circuits
  via nodal analysis, 50–51
  via voltage division, 51
Simulation, with LabVIEW program, 858–859
Simulation palette, in LabVIEW program, 881–882
Simulation parameters, in MATLAB Simulink program, 846–847
Simulink (MATLAB), 190, 196, 199, 225, 228, 247, 297, 335, 384, 720–721, 758, 787–835
Single loop
  via differential equation, 48–49
  via mesh analysis, 48–50
  via transform methods, 50
Single node, 50–51
Sinha, N. K., 721
Sinusoidal frequency analysis, 5
Sinusoidal inputs, 19
SISO Design Tool, 20
Sivan, R., 300
Skewis, T., 299
Smith, C. A., 532, 662
Smith, C. L., 781
Smoother, block diagram, 440
Soft Arm, 616
Solar alpha rotary joints, 380
Solar arrays, in space station, 380
Soleimani-Mosheni, M., 337, 624
Solenoid coil circuit, 289
Solomonow, M., 234
Son, M., 662
Source shaft, 76
Space shuttle, 6
  automatic steering program, 439
  main engine controller, 10
  multiple subsystems in, 237
  pitch control system, 287–288
Space station
  solar arrays, 380
  vibration stabilization, 440
Sparks, A. W., 775, 781
Speed control, 4–5
Sperry Gyroscope Company, 5
Spilman, D., 531, 721
s-plane, digital system stability via, 747–748
s-plane mapping, onto *z*-plane, 743
Split-power hybrid-electric vehicle, 30
Spring constant, 62, 70
Spruijt, J. N., 234, 722
Squid jet locomotion, 109

Stability, 11
closed-loop poles/response in, 304–305
definition, 302
determining, 576
evaluation of, 303
with MATLAB program, 804–805
Maxwell's criterion of, 5
of missile control system, 743–745
range of sampling interval for, 745–746
Routh-Hurwitz criterion of, 305–308
in state space, 320–323
via epsilon method, 308–309
via mapping only positive $j\omega$-axis, 571–573
via Nyquist diagram, 569–573
via reverse coefficients, 310
via Routh-Hurwitz criterion, 318–319
via Routh table, 310–311
Stability design/transient performance, antenna control and, 606–607
Stability design via gain
in antenna control, 323–324
in UFSS, 324
Stability design via root locus, in digital control systems, 755–756
Stability/transient response design, via gain adjustment, 626
Stable systems, steady-state error and, 341
Stall torque, 82
State equations, 122, 123, 124
Laplace transform solution of, 198–202
signal-flow graphs of, 254–256
similarity transformations on, 267
State-feedback design, 683
State solutions, time domain solution of, 203–212
State space, 123
alternative representations in, 256–265
cascade form of, 257–259
controller canonical form of, 260–262
diagonalizing system in, 270–271
graphic representation of, 123
observer canonical form of, 262–265
parallel form of, 259–260
stability in, 320–323
transfer function conversion to, 132–138
State space conversion, to transfer function, 139–141
State-space design, 663–665
for antenna azimuth position control, 708
State-space representation, 18, 117
advantages of, 119
antenna control and, 144
application of, 124–132
computer simulation and, 149
general, 123–124
linearization and, 141–143
of plant, 666
to transfer function, 140
State-transition matrix, 203
Laplace transform of, 203–204
via Laplace transform, 206–207
State-variable feedback, plant with, 706
State variables, 123
linearly independent, 124–125
minimum number of, 125–126
State vector, 123
graphic representation of, 123
Static error constants, 349–350, 464
from Bode plots, 595–596
steady-state error via, 350–352
Steady-state actuating signal, for nonunity feedback systems, 361
Steady-state error design
via cascade compensation, 626
via integral control, 700–704
Steady-state errors, 2, 15, 19. *See also* Errors
antenna control and, 368–371
assembly-line robots and, 354
definition, 340
digital control systems and, 749–753
digital feedback control system for, 750
for disturbances, 356–358
evaluating, 341–342
finding, 752
from frequency response, 593–597
improving, 457–458, 482–494
with MATLAB program, 805–806
for no integration systems, 347–348
for nonunity feedback systems, 358–362
for one integration systems, 348–349
parabolic input and, 347
ramp input and, 341, 342, 346–347
sensitivity of, 362–364
sources of, 343
specifications, 353–356
stable systems and, 341
from step disturbances, 357
step input and, 346
for systems in state space, 364–367
system type and, 352–353
in terms of $G(s)$, 345–348
in terms of $T(s)$, 344–345
test inputs, 340–341
test waveforms for, 340
transient response, 19, 482–494
for unity feedback systems, 343–349
using final value theorem, 365
using input substitution, 367
via cascade compensation, 459–469
via static error constants, 350–352

Steady-state response, 11
Steady-state velocity error, 375
Steam-driven power generators, 524
Steam generator, water level control in, 333
Steam pressure control, 4
Steel production, continuous casting in, 294
Steering control, of F4-E military aircraft, 153
Stefani, R. T., 32, 220, 233, 299, 337, 380, 386, 531, 623, 662, 721
Stefanides, E. J., 662
Step, 19
Step disturbances, steady-state errors from, 357
Step input, steady state error and, 346
Step responses
  computer simulation of, 429
  of gain-adjusted antenna control system, 427
  of nonminimum-phase network, 194
  of nonminimum-phase system, 192
  of pitch control loop without rate feedback, 429
  of pitch control loop with rate feedback, 430
  for second-order system damping cases, 172
  of second-order underdamped systems, 184
  of three-pole systems, 189
  for transfer functions, 190
Step response simulation, for minor-loop feedback compensation, 503
s-transform, table of, 732
Strobel, K. L., 221, 623
Student population control system, 25
Submarine autopilot, 154
Subsystems, 2. *See also* Multiple subsystems
Summers, T. A., 233
Summing admittances, 56, 86
Summing impedances, 56, 57
Summing junctions, 8
  block diagram for, 241
Summing torques, of pendulum, 142
Summing voltages, 57
Sun, M., 225, 233
Superposition, 88
Susceptance, 54n
Suspension design, for quarter-car model, 107
Sweet, L. M., 233, 623
Symbolic Math Toolbox, 21
Symbols list, 783
Symmetry, root locus sketching and, 397–398
System matrix, diagonalizing a, 268
System representation, block diagram of, 34
System response, 162–163
  with additional poles, 186–190
  from transfer function, 46
  with zeros, 191–196
Systems in state space, steady-state errors for, 364–367
System step response test, laboratory results of, 168
System type, steady-state errors and, 352–353
System variable, 123

T(s), steady state error as, 344–345
Tabular method (Raible), 745
Tachometer
  as position control system, 495
  transfer function of, 496
Tadeo, F., 721
Tanis, D., 287
Tan, X., 152, 160
Tarín, C., 153, 160, 715, 722
Tactile feedback suit, 27–28
Target environment, robotic manipulator and, 153
Tasch, U., 292, 300, 776, 781
Taylor series, 58, 59, 80, 142
Television imaging system, robot with, 152
Temperature control system, 4, 24
  chemical process, 525
Testing
  first-order transfer functions via, 167–168
  second-order transfer functions via, 186
Test inputs, for steady-state error, 340–341
Test waveforms, 19
  for steady-state error, 340
Teufel, E., 160, 722
Theophylline concentration, 221
Thermistor, 9
Thermostat, 6
Third-order observer, 684
Third-order system gain design, 416–419
  system characteristics, 417–418
Thomas, B., 333, 337, 532, 616, 624
Three-loop electrical network, 56
Three-mode controllers. *See* PID (proportional-plus-integral-plus-derivative) controllers
Three-pole systems
  comparing responses of, 189–190
  component responses of, 188
  step responses of, 189
Thrust vectoring, 287
Thukral, A., 337, 661
Tie, B., 109, 116
Time constant, 166–167
  exponential, 174
  reciprocal of, 167
Time delay systems, 597–602
  frequency response plots of, 598–599
  percent overshoot for, 600–601
  range of gain for stability for, 599–600
Time domain modeling, 117–122
  with MATLAB program, 793–795
  of state solutions, 203–212

Time function
  Laplace transform of, 35–36
  z-transform of, 731–732
Time response
  with MATLAB program, 795–799
  nonlinearities and, 196–198
Time-varying systems, 118
Timothy, L. K., 160, 233, 300, 337, 722
Titanic, 272
Top-down design, 22
Torque, of motor, 80
Torque-angular displacement, 70
Torque-angular velocity, 70
Torque equivalent mechanical loading, 80
Torque-speed curve, 82, 83
  with armature voltage, 82
Total response, 11
Tou, J., 781
Towed vehicle roll control, 332
Tower Trainer 60 Unmanned Aerial Vehicle, 657
Towsley, D., 386, 453, 623, 662
Tracking radar, 382
Train stopping, closed-loop vehicle response for, 441
Transducer
  inner ear hair cell, 105
  input, 8
  output, 9
Transfer function(s), 17
  block diagram of, 45
  from Bode plots, 603–605
  decomposing, 136, 137
  for differential equation, 45
  of digital control systems, 735–739
  electrical network, 47–61
  experimentally obtaining, 602–605
  frequency response from, 538–539
  of human leg, 95–96
  matrix, 139
  nonminimum-phase system of, 192–194
  peak time from, 181–182
  percent overshoot from, 181–182
  poles of, 163
  rise time from, 181–182
  sampled-data systems and, 735–739
  second-order approximation, 190
  settling time from, 181–182
  state space conversion to, 140
  state-space representation to, 140
  step responses for, 190
  system response from, 46
  of translational mechanical system, 61–69
  via Mason's rule, 252–253
  zeros of, 163
Transfer function conversion
  with constant term in numerator, 134–135
  with polynomial in numerator, 136–137
  to state space, 132–138
Transfer function poles, eigenvalues and, 200–202
Transformation
  controller design by, 679–681
  observer design by, 695–697
Transformation matrix, 266, 268, 270, 272
Transformations
  bilinear, 746–747
  to canonical form, 260–263
  controller design by, 678–681
  observer design by, 693–697
  similarity, 266–270
Transformed circuit, 49
Transformed free-body diagram, 63
Transform methods, single loop via, 50
Transform of the response, 177
Transient response, 2, 10
  through component design, 185–186
  desired, 10
  gain design for, 247–248
  improving, 456–457
  modeling, 19
  real-axis pole and, 165
  steady-state errors and, 19, 482–494
  via gain adjustment, 627–630
Transient response design, via gain adjustment, 415–419, 626
Transient response design via gain adjustment
  antenna control and, 426–427, 765–767
  digital control systems and, 756–757
  UFSS and, 427–430
Transient response improvement, via cascade compensation, 469–481
Transient response modeling steady-state error, 19
Transient response on z-plane, digital control systems and, 753–755
Transition method, 303
Translational mechanical system
  representation of, 130–131
  transfer functions, 61–69
Transpose, 128
A Treatise on the Stability of a Given State of Motion (Routh), 5
TryIt, 40-43, 46, 57, 72, 138, 141, 186, 190, 191, 195, 202, 245, 248, 261, 263, 271, 308, 316, 322, 353, 355, 359, 364, 365, 391, 397, 414, 467, 477, 559, 570, 576, 579, 588, 601, 630, 634, 641, 672, 676, 689, 693, 738, 758
Tumor cell growth model, 109
Turnbull, G. A., 32
Tustin transformation, 759
Twizell, E. H., 115
Two degrees of freedom translational mechanical system, 65–66
Two-lop electrical network, 52–53
Two-pole system, zeros and, 191
Two-tank liquid control system, 521

Tyberg, V. J., 232
Tyner, M., 722

UFSS. *See* Unmanned Free-Swimming Submersible
UNAIDS, 32
Unbounded input, 302
Uncompensated/compensated system, of ideal derivative compensation, 472–473
Uncompensated system
  ideal derivative compensation and, 472
  root locus for, 462
Uncompensated system response, ideal integral compensated system response and, 463
Uncontrollable system, 673
Undamped response, 171, 172
Undamped second-order systems, 177–186
Undamped sinusoidal oscillations, 213
Underdamped curve, 174
Underdamped response, 170, 172
  using poles, 171
Underdamped second-order system, pole plot for, 182
Underwater remote-controlled vehicle, 317
Uniform-rate sampling, 728
Uniform rectangular pulse train, 729
Unit multiplicity, 312n
Unit parabolic input, for digital feedback control system, 751
Unit ramp input, for digital feedback control system, 751
Unit step, first-order system and, 166
Unit step input, for digital feedback control system, 750–751
Unity feedback systems
  forming an equivalent, 358
  steady state error for, 343–349
Unity gain, 9
Unmanned autonomous vehicle, 527–528
Unmanned Free-Swimming Submersible (UFSS), 162, 272
  lead/feedback compensation in, 511–513
  open-loop pitch response and, 207–208
  pitch angle control representation, 276–277
  pitch control loop for, 211
  stability design via gain in, 324
  transient design via gain and, 427–430
Unobservability, via observability matrix, 692–693
Unstable systems, 303. *See also* Stability
Ünyelioglu, K. A., 337, 454

Vacuum robot, 220
van der Helm, F. C. T., 292, 299
van der Molen, G. M., 154, 160
Van de Vegte, J., 532
Van Dijk, E., 225, 234, 722
Van Valkenburg, M. E., 116
Variable speed wind turbine, feedback control, 718
Vaughan, N. D., 26, 32, 289, 300
Vector-matrix form, 127
Vector representation, of complex numbers, 389–391
Vectors
  evaluation of complex function via, 391
  Laplace transform of, 139n
  state, 123
Vehicle steering control model, 441
Velocity, 374
Velocity constant, 594
Vertical spindle surface grinding, 333
Venter, J. W., 31, 110, 115, 159, 232, 299, 337, 386, 455, 531, 623, 662, 721, 789
Verde, L., 153, 159, 299, 452
Vertical risers, 4
Viaro, U., 290, 300, 386
Vibration stabilization, in space station, 440
Video disc laser recording, 370
Vidyasagar, M., 116, 386, 453
Vieira, A., 160, 721
Virkkunen, J., 108, 116, 337, 623
Virtual Experiment, 65, 83, 142, 166, 182, 247, 313, 357, 473, 501
Virtual reality simulator, 293
Viscous damper, 64, 125
Viscous damping, 17
Visual cue, human response to, 220
Voltage-charge, 48
Voltage-current, 48
Voltage-dependent current source, 128
Voltage division, simple circuits via, 51
Voltage droop control, 442

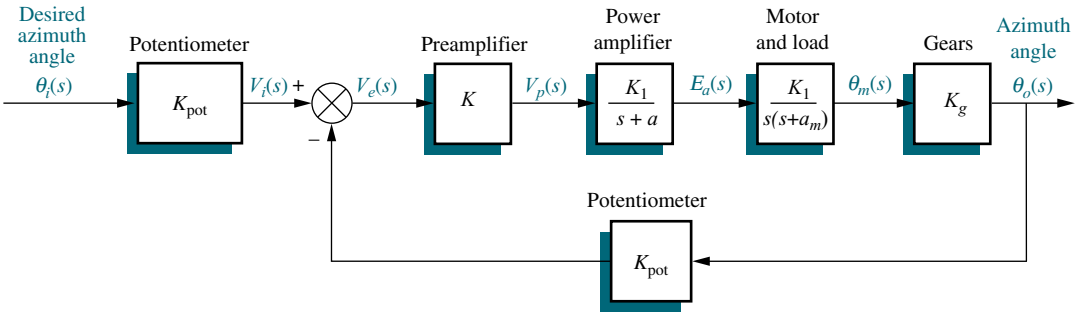
Walking robots. *See* Robots
Wang, H., 386, 453, 623
Wang, J. Z., 109, 116
Water clock, 4
Water level control, in steam generator, 333
Watt, James, 5
Waveforms, 19
Weiss, R., 152, 160
Welkowitz, W., 109, 116
Wie, B., 624
Winders, 25
Windmill speed control, 4–5
Wind turbines, 446
Wittenmark, B., 759, 780
Wolfson, P., 233
Wong, M. C., 386, 453
Woods Hole Oceanographic Institution, 317
Wormly, D. N., 28, 32, 110, 116, 160, 233, 300, 337, 386, 453, 532, 624, 662, 721

Xia, X., 31, 110, 115, 159, 232, 299, 337, 386, 452, 531, 623, 662, 721, 780
Xiong, Y., 233

Yan, T., 330, 338, 447, 454, 662
Yaniv, Y., 292, 300
Yingst, J. C., 780
Yoneyama, T., 116, 453, 623

Zbinden, A. M., 32
Zedka, M., 223, 234
Zero(s), 162–163
  entire row is, 310–311
  in first column, 308–309, 314–315
  of first-order system, 163
  in Routh table, 308–311
  system response with, 191–196
  of transfer function, 163
  two-pole system and, 191
Zero-input response, 203
Zero-order hold
  ideal sampling and, 730
  modeling, 729–730
Zero-order sample-and-hold, 726, 730
Zero-state response, 203, 214
Zhou, B. H., 221, 234
z-plane
  digital system stability via, 742–743
  s-plane mapping onto, 743
z-transform, 728, 730–735
  digital control systems and, 730–735
  sampled-data systems and, 739
  table of, 732
  theorems, 733
  of time function, 731–732

Block Diagram



Schematic Parameters

Parameter	Configuration 1	Configuration 2	Configuration 3
V	10	10	10
n	10	1	1
K	—	—	—
K_1	100	150	100
a	100	150	100
R_a	8	5	5
J_a	0.02	0.05	0.05
D_a	0.01	0.01	0.01
K_b	0.5	1	1
K_t	0.5	1	1
N_1	25	50	50
N_2	250	250	250
N_3	250	250	250
J_L	1	5	5
D_L	1	3	3

Block Diagram Parameters

Parameter	Configuration 1	Configuration 2	Configuration 3
K_{pot}	0.318		
K	—		
K_1	100		
a	100		
K_m	2.083		
a_m	1.71		
K_g	0.1		

Note: reader may fill in Configuration 2 and Configuration 3 columns after completing the antenna control Case Study challenge problems in Chapters 2 and 10, respectively.

Key Equations

Modeling

$$\frac{V_o(s)}{V_i(s)} = -\frac{Z_2(s)}{Z_1(s)} \quad (2.97); \qquad \frac{V_o(s)}{V_i(s)} = \frac{Z_1(s) + Z_2(s)}{Z_1(s)} \quad (2.104)$$

$$\frac{\theta_2}{\theta_1} = \frac{r_1}{r_2} = \frac{N_1}{N_2} \quad (2.133); \qquad \frac{T_2}{T_1} = \frac{\theta_1}{\theta_2} = \frac{N_2}{N_1} \quad (2.135)$$

$$\left(\frac{\text{Number of teeth of gear on } \textit{destination} \text{ shaft}}{\text{Number of teeth of gear on } \textit{source} \text{ shaft}} \right)^2 \quad \text{(see after 2.138)}$$

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/(R_a J_m)}{s\left[s + \frac{1}{J_m}\left(D_m + \frac{K_t K_b}{R_a}\right)\right]} \quad (2.153)$$

$$\frac{K_t}{R_a} = \frac{T_{\text{stall}}}{e_a} \quad (2.162); \quad K_b = \frac{e_a}{\omega_{\text{no-load}}} \quad (2.163)$$

$$T(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (3.73)$$

Time Response

$$T_r = \frac{2.2}{a} \quad (4.9); \quad T_s = \frac{4}{a} \quad (4.10)$$

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.22)$$

$$\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100 \quad (4.38)$$

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \quad (4.39)$$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \quad (4.34); \quad T_s = \frac{4}{\zeta\omega_n} \quad (4.42)$$

Steady-State Error

$$e(\infty) = e_{\text{step}}(\infty) = \frac{1}{1 + \lim_{s\to 0} G(s)} \quad (7.30); \quad K_p = \lim_{s\to 0} G(s) \quad (7.33)$$

$$e(\infty) = e_{\text{ramp}}(\infty) = \frac{1}{\lim_{s\to 0} sG(s)} \quad (7.31); \quad K_v = \lim_{s\to 0} sG(s) \quad (7.34)$$

$$e(\infty) = e_{\text{parabola}}(\infty) = \frac{1}{\lim_{s\to 0} s^2 G(s)} \quad (7.32); \quad K_a = \lim_{s\to 0} s^2 G(s) \quad (7.35)$$

Root Locus

$$\angle KG(s)H(s) = -1 = 1\angle(2k+1)180^\circ \quad (8.13)$$

$$\sigma_a = \frac{\sum \text{ finite poles} - \sum \text{ finite zeros}}{\#\text{ finite poles} - \#\text{ finite zeros}} \quad (8.27)$$

$$\theta_a = \frac{(2k+1)\pi}{\#\text{ finite poles} - \#\text{ finite zeros}} \quad (8.28)$$

$$\theta = \sum \text{finite zero angles} - \sum \text{finite pole angles}$$

$$K = \frac{1}{|G(s)H(s)|} = \frac{1}{M} = \frac{\prod \text{ finite pole lengths}}{\prod \text{ finite zero lengths}} \quad (8.51)$$

Frequency Response

$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (10.52); \quad \omega_p = \omega_n\sqrt{1-2\zeta^2} \quad (10.53)$$

$$\omega_{\text{BW}} = \omega_n\sqrt{(1-2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} \quad (10.54)$$

$$\Phi_M = \tan^{-1}\frac{2\zeta}{\sqrt{-2\zeta^2 + \sqrt{1+4\zeta^4}}} \quad (10.73)$$

$$\phi_{\max} = \tan^{-1}\frac{1-\beta}{2\sqrt{\beta}} = \sin^{-1}\frac{1-\beta}{1+\beta} \quad (11.11)$$

$$\omega_{\max} = \frac{1}{T\sqrt{\beta}} \quad (11.9); \quad |G_c(j\omega_{\max})| = \frac{1}{\sqrt{\beta}} \quad (11.12)$$

State Space

$$\mathbf{C_M} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (12.26)$$

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{B}r; \quad y = \mathbf{Cx} \quad (12.3); \quad \mathbf{O_M} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (12.79)$$

$$\dot{\mathbf{e}}_\mathbf{x} = (\mathbf{A} - \mathbf{LC})\mathbf{e}_\mathbf{x}; \quad y - \hat{y} = \mathbf{Ce}_\mathbf{x} \quad (12.64)$$

Digital Control

$$e^*(\infty) = \lim_{z\to 1}(1 - z^{-1})E(z) \quad (13.66)$$

$$K_p = \lim_{z\to 1} G(z) \quad (13.70); \quad K_v = \frac{1}{T}\lim_{z\to 1}(z-1)G(z) \quad (13.73)$$

$$K_a = \frac{1}{T^2}\lim_{z\to 1}(z-1)^2 G(z) \quad (13.75)$$

Solutions to Skill-Assessment Exercises

CHAPTER 2

2.1

The Laplace transform of t is $\frac{1}{s^2}$ using Table 2.1, Item 3. Using Table 2.2, Item 4, $F(s) = \frac{1}{(s+5)^2}$.

2.2

Expanding $F(s)$ by partial fractions yields:

$$F(s) = \frac{A}{s} + \frac{B}{s+2} + \frac{C}{(s+3)^2} + \frac{D}{(s+3)}$$

where,

$$A = \left.\frac{10}{(s+2)(s+3)^2}\right|_{S\to 0} = \frac{5}{9} \quad B = \left.\frac{10}{s(s+3)^2}\right|_{S\to -2} = -5$$

$$C = \left.\frac{10}{s(s+2)}\right|_{S\to -3} = \frac{10}{3}, \text{ and } D = (s+3)^2\left.\frac{dF(s)}{ds}\right|_{s\to -3} = \frac{40}{9}$$

Taking the inverse Laplace transform yields,

$$f(t) = \frac{5}{9} - 5e^{-2t} + \frac{10}{3}te^{-3t} + \frac{40}{9}e^{-3t}$$

2.3

Taking the Laplace transform of the differential equation assuming zero initial conditions yields:

$$s^3C(s) + 3s^2C(s) + 7sC(s) + 5C(s) = s^2R(s) + 4sR(s) + 3R(s)$$

Collecting terms,

$$(s^3 + 3s^2 + 7s + 5)C(s) = (s^2 + 4s + 3)R(s)$$

Thus,

$$\frac{C(s)}{R(s)} = \frac{s^2 + 4s + 3}{s^3 + 3s^2 + 7s + 5}$$

2.4

$$G(s) = \frac{C(s)}{R(s)} = \frac{2s+1}{s^2+6s+2}$$

Cross multiplying yields,

$$\frac{d^2c}{dt^2} + 6\frac{dc}{dt} + 2c = 2\frac{dr}{dt} + r$$

2.5

$$C(s) = R(s)G(s) = \frac{1}{s^2} * \frac{s}{(s+4)(s+8)} = \frac{1}{s(s+4)(s+8)} = \frac{A}{s} + \frac{B}{(s+4)} + \frac{C}{(s+8)}$$

where

$$A = \left.\frac{1}{(s+4)(s+8)}\right|_{S\to 0} = \frac{1}{32} \quad B = \left.\frac{1}{s(s+8)}\right|_{S\to -4} = -\frac{1}{16}, \text{ and } C = \left.\frac{1}{s(s+4)}\right|_{S\to -8} = \frac{1}{32}$$

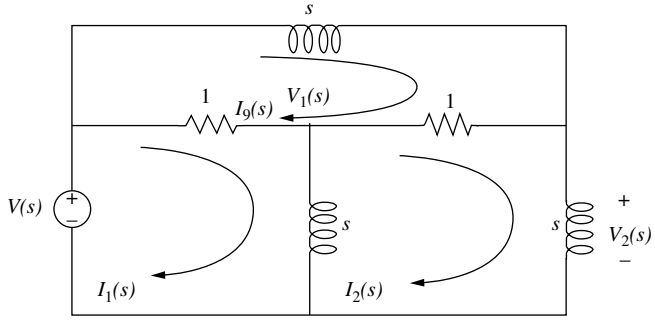
Thus,

$$c(t) = \frac{1}{32} - \frac{1}{16}e^{-4t} + \frac{1}{32}e^{-8t}$$

2.6

Mesh Analysis

Transforming the network yields,



Now, writing the mesh equations,

$$\begin{aligned}(s+1)I_1(s) - sI_2(s) - I_3(s) &= V(s)\\ -sI_1(s) + (2s+1)I_2(s) - I_3(s) &= 0\\ -I_1(s) - I_2(s) + (s+2)I_3(s) &= 0\end{aligned}$$

Solving the mesh equations for $I_2(s)$,

$$I_2(s) = \frac{\begin{vmatrix}(s+1) & V(s) & -1\\ -s & 0 & -1\\ -1 & 0 & (s+2)\end{vmatrix}}{\begin{vmatrix}(s+1) & -s & -1\\ -s & (2s+1) & -1\\ -1 & -1 & (s+2)\end{vmatrix}} = \frac{(s^2+2s+1)V(s)}{s(s^2+5s+2)}$$

But, $V_L(s) = sI_2(s)$

Hence,

$$V_L(s) = \frac{(s^2 + 2s + 1)V(s)}{(s^2 + 5s + 2)}$$

or

$$\frac{V_L(s)}{V(s)} = \frac{s^2 + 2s + 1}{s^2 + 5s + 2}$$

Nodal Analysis

Writing the nodal equations,

$$\left(\frac{1}{s} + 2\right)V_1(s) - V_L(s) = V(s)$$
$$-V_1(s) + \left(\frac{2}{s} + 1\right)V_L(s) = \frac{1}{s}V(s)$$

Solving for $V_L(s)$,

$$V_L(s) = \frac{\begin{vmatrix} \left(\frac{1}{s} + 2\right) & V(s) \\ -1 & \frac{1}{s}V(s) \end{vmatrix}}{\begin{vmatrix} \left(\frac{1}{s} + 2\right) & -1 \\ -1 & \left(\frac{2}{s} + 1\right) \end{vmatrix}} = \frac{(s^2 + 2s + 1)V(s)}{(s^2 + 5s + 2)}$$

or

$$\frac{V_L(s)}{V(s)} = \frac{s^2 + 2s + 1}{s^2 + 5s + 2}$$

2.7

Inverting

$$G(s) = -\frac{Z_2(s)}{Z_1(s)} = \frac{-100000}{\left(10^5/s\right)} = -s$$

Noninverting

$$G(s) = \frac{[Z_1(s) + Z(s)]}{Z_1(s)} = \frac{\left(\frac{10^5}{s} + 10^5\right)}{\left(\frac{10^5}{s}\right)} = s + 1$$

2.8

Writing the equations of motion,

$$(s^2 + 3s + 1)X_1(s) - (3s + 1)X_2(s) = F(s)$$
$$-(3s + 1)X_1(s) + (s^2 + 4s + 1)X_2(s) = 0$$

Solving for $X_2(s)$,

$$X_2(s) = \frac{\begin{vmatrix} (s^2+3s+1) & F(s) \\ -(3s+1) & 0 \end{vmatrix}}{\begin{vmatrix} (s^2+3s+1) & -(3s+1) \\ -(3s+1) & (s^2+4s+1) \end{vmatrix}} = \frac{(3s+1)F(s)}{s(s^3+7s^2+5s+1)}$$

Hence,

$$\frac{X_2(s)}{F(s)} = \frac{(3s+1)}{s(s^3+7s^2+5s+1)}$$

2.9

Writing the equations of motion,

$$(s^2+s+1)\theta_1(s) - (s+1)\theta_2(s) = T(s)$$
$$-(s+1)\theta_1(s) + (2s+2)\theta_2(s) = 0$$

where $\theta_1(s)$ is the angular displacement of the inertia.

Solving for $\theta_2(s)$,

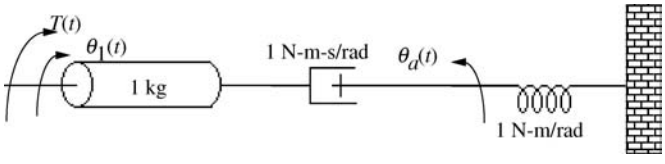
$$\theta_2(s) = \frac{\begin{vmatrix} (s^2+s+1) & T(s) \\ -(s+1) & 0 \end{vmatrix}}{\begin{vmatrix} (s^2+s+1) & -(s+1) \\ -(s+1) & (2s+2) \end{vmatrix}} = \frac{(s+1)F(s)}{2s^3+3s^2+2s+1}$$

From which, after simplification,

$$\theta_2(s) = \frac{1}{2s^2+s+1}$$

2.10

Transforming the network to one without gears by reflecting the 4 N-m/rad spring to the left and multiplying by $(25/50)^2$, we obtain,



Writing the equations of motion,

$$(s^2+s)\theta_1(s) - s\theta_a(s) = T(s)$$
$$-s\theta_1(s) + (s+1)\theta_a(s) = 0$$

where $\theta_1(s)$ is the angular displacement of the 1-kg inertia.

Solving for $\theta_a(s)$,

$$\theta_a(s) = \frac{\begin{vmatrix} (s^2+s) & T(s) \\ -s & 0 \end{vmatrix}}{\begin{vmatrix} (s^2+s) & -s \\ -s & (s+1) \end{vmatrix}} = \frac{sT(s)}{s^3+s^2+s}$$

From which,

$$\frac{\theta_a(s)}{T(s)} = \frac{1}{s^2 + s + 1}$$

But, $\theta_2(s) = \frac{1}{2}\theta_a(s)$.

Thus,

$$\frac{\theta_2(s)}{T(s)} = \frac{1/2}{s^2 + s + 1}$$

2.11

First find the mechanical constants.

$$J_m = J_a + J_L\left(\frac{1}{5} * \frac{1}{4}\right)^2 = 1 + 400\left(\frac{1}{400}\right) = 2$$

$$D_m = D_a + D_L\left(\frac{1}{5} * \frac{1}{4}\right)^2 = 5 + 800\left(\frac{1}{400}\right) = 7$$

Now find the electrical constants. From the torque-speed equation, set $\omega_{\mathrm{m}} = 0$ to find stall torque and set $T_m = 0$ to find no-load speed. Hence,

$$T_{stall} = 200$$
$$\omega_{no-load} = 25$$

which,

$$\frac{K_t}{R_a} = \frac{T_{stall}}{E_a} = \frac{200}{100} = 2$$
$$K_b = \frac{E_a}{\omega_{no-load}} = \frac{100}{25} = 4$$

Substituting all values into the motor transfer function,

$$\frac{\theta_m(s)}{E_a(s)} = \frac{\frac{K_T}{R_a J_m}}{s\left(s + \frac{1}{J_m}\right)\left(D_m + \frac{K_T K_b}{R_a}\right)} = \frac{1}{s\left(s + \frac{15}{2}\right)}$$

where $\theta_m(s)$ is the angular displacement of the armature.

Now $\theta_L(s) = \frac{1}{20}\theta_m(s)$. Thus,

$$\frac{\theta_L(s)}{E_a(s)} = \frac{1/20}{s\left(s + \frac{15}{2}\right)}$$

2.12

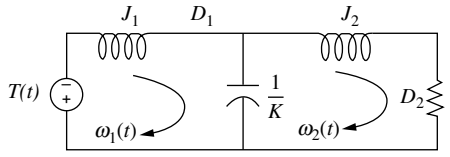
Letting

$$\theta_1(s) = \omega_1(s)/s$$
$$\theta_2(s) = \omega_2(s)/s$$

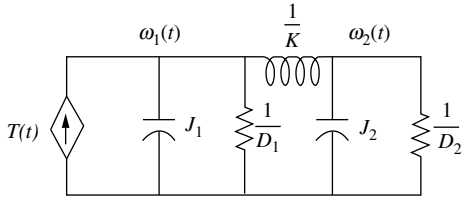
in Eqs. 2.127, we obtain

$$\left(J_1 s + D_1 + \frac{K}{s}\right)\omega_1(s) - \frac{K}{s}\omega_2(s) = T(s)$$
$$-\frac{K}{s}\omega_1(s) + \left(J_2 s + D_2 + \frac{K}{s}\right)\omega_2(s)$$

From these equations we can draw both series and parallel analogs by considering these to be mesh or nodal equations, respectively.



Series analog



Parallel analog

2.13

Writing the nodal equation,

$$C\frac{dv}{dt} + i_r - 2 = i(t)$$

But,

$$C = 1$$
$$v = v_o + \delta v$$
$$i_r = e^{v_r} = e^{v} = e^{v_o + \delta v}$$

Substituting these relationships into the differential equation,

$$\frac{d(v_o + \delta v)}{dt} + e^{v_o + \delta v} - 2 = i(t) \tag{1}$$

We now linearize e^v.

The general form is

$$f(v) - f(v_o) \approx \left.\frac{df}{dv}\right|_{v_o} \delta v$$

Substituting the function, $f(v) = e^v$, with $v = v_o + \delta v$ yields,

$$e^{v_o + \delta v} - e^{v_o} \approx \left.\frac{de^v}{dv}\right|_{v_o} \delta v$$

Solving for $e^{v_o+\delta v}$,

$$e^{v_o+\delta v} = e^{v_o} + \frac{de^v}{dv}\bigg|_{v_o} \delta v = e^{v_o} + e^{v_o}\delta v$$

Substituting into Eq. (1)

$$\frac{d\delta v}{dt} + e^{v_o} + e^{v_o}\delta v - 2 = i(t) \tag{2}$$

Setting $i(t) = 0$ and letting the circuit reach steady state, the capacitor acts like an open circuit. Thus, $v_o = v_r$ with $i_r = 2$. But, $i_r = e^{v_r}$ or $v_r = \ln i_r$.

Hence, $v_o = \ln 2 = 0.693$. Substituting this value of v_o into Eq. (2) yields

$$\frac{d\delta v}{dt} + 2\delta v = i(t)$$

Taking the Laplace transform,

$$(s+2)\delta v(s) = I(s)$$

Solving for the transfer function, we obtain

$$\frac{\delta v(s)}{I(s)} = \frac{1}{s+2}$$

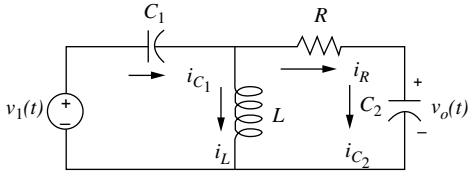
or

$\dfrac{V(s)}{I(s)} = \dfrac{1}{s+2}$ about equilibrium.

CHAPTER 3

3.1

Identifying appropriate variables on the circuit yields



Writing the derivative relations

$$\begin{aligned} C_1\frac{dv_{C_1}}{dt} &= i_{C_1} \\ L\frac{di_L}{dt} &= v_L \\ C_2\frac{dv_{C_2}}{dt} &= i_{C_2} \end{aligned} \tag{1}$$

Using Kirchhoff's current and voltage laws,

$$\begin{aligned} i_{C_1} &= i_L + i_R = i_L + \frac{1}{R}(v_L - v_{C_2}) \\ v_L &= -v_{C_1} + v_i \\ i_{C_2} &= i_R = \frac{1}{R}(v_L - v_{C_2}) \end{aligned}$$

Substituting these relationships into Eqs. (1) and simplifying yields the state equations as

$$\begin{aligned} \frac{dv_{C_1}}{dt} &= -\frac{1}{RC_1}v_{C_1} + \frac{1}{C_1}i_L - \frac{1}{RC_1}v_{C_2} + \frac{1}{RC_1}v_i \\ \frac{di_L}{dt} &= -\frac{1}{L}v_{C_1} + \frac{1}{L}v_i \\ \frac{dv_{C_2}}{dt} &= -\frac{1}{RC_2}v_{C_1} - \frac{1}{RC_2}v_{C_2}\frac{1}{RC_2}v_i \end{aligned}$$

where the output equation is

$$v_o = v_{C_2}$$

Putting the equations in vector-matrix form,

$$\dot{\mathbf{x}} = \begin{bmatrix} -\frac{1}{RC_1} & \frac{1}{C_1} & -\frac{1}{RC_1} \\ -\frac{1}{L} & 0 & 0 \\ -\frac{1}{RC_2} & 0 & -\frac{1}{RC_2} \end{bmatrix}\mathbf{x} + \begin{bmatrix} \frac{1}{RC_1} \\ \frac{1}{L} \\ \frac{1}{RC_2} \end{bmatrix} v_i(t)$$

$$y = [\,0 \quad 0 \quad 1\,]\mathbf{x}$$

3.2

Writing the equations of motion

$$\begin{aligned} (s^2 + s + 1)X_1(s) \qquad -sX_2(s) \qquad\qquad &= F(s) \\ -sX_1(s) + (s^2 + s + 1)X_2(s) \qquad -X_3(s) \quad &= 0 \\ -X_2(s) + (s^2 + s + 1)X_3(s) &= 0 \end{aligned}$$

Taking the inverse Laplace transform and simplifying,

$$\begin{aligned} \ddot{x}_1 &= -\dot{x}_1 - x_1 + \dot{x}_2 + f \\ \ddot{x}_2 &= \dot{x}_1 - \dot{x}_2 - x_2 + x_3 \\ \ddot{x}_3 &= -\dot{x}_3 - x_3 + x_2 \end{aligned}$$

Defining state variables, z_i,

$$z_1 = x_1; z_2 = \dot{x}_1; z_3 = x_2; z_4 = \dot{x}_2; z_5 = x_3; z_6 = \dot{x}_3$$

Writing the state equations using the definition of the state variables and the inverse transform of the differential equation,

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= \ddot{x}_1 = -\dot{x}_1 - x_1 + \dot{x}_2 + f = -z_2 - z_1 + z_4 + f \\
\dot{z}_3 &= \dot{x}_2 = z_4 \\
\dot{z}_4 &= \ddot{x}_2 = \dot{x}_1 - \dot{x}_2 - x_2 + x_3 = z_2 - z_4 - z_3 + z_5 \\
\dot{z}_5 &= \dot{x}_3 = z_6 \\
\dot{z}_6 &= \ddot{x}_3 = -\dot{x}_3 - x_3 + x_2 = -z_6 - z_5 + z_3
\end{aligned}$$

The output is z_5. Hence, $y = z_5$. In vector-matrix form,

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} f(t); y = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{z}$$

3.3

First derive the state equations for the transfer function without zeros.

$$\frac{X(s)}{R(s)} = \frac{1}{s^2 + 7s + 9}$$

Cross multiplying yields

$$\left(s^2 + 7s + 9\right) X(s) = R(s)$$

Taking the inverse Laplace transform assuming zero initial conditions, we get

$$\ddot{x} + 7\dot{x} + 9x = r$$

Defining the state variables as,

$$\begin{aligned}
x_1 &= x \\
x_2 &= \dot{x}
\end{aligned}$$

Hence,

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \ddot{x} = -7\dot{x} - 9x + r = -9x_1 - 7x_2 + r
\end{aligned}$$

Using the zeros of the transfer function, we find the output equation to be,

$$c = 2\dot{x} + x = x_1 + 2x_2$$

Putting all equation in vector-matrix form yields,

$$\begin{aligned}
\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -9 & -7 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \\
c &= \begin{bmatrix} 1 & 2 \end{bmatrix} \mathbf{x}
\end{aligned}$$

3.4

The state equation is converted to a transfer function using

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \tag{1}$$

where

$$\mathbf{A} = \begin{bmatrix} -4 & -1.5 \\ 4 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \text{ and } \mathbf{C} = [\,1.5 \quad 0.625\,].$$

Evaluating $(s\mathbf{I} - \mathbf{A})$ yields

$$(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s+4 & 1.5 \\ -4 & s \end{bmatrix}$$

Taking the inverse we obtain

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{s^2 + 4s + 6}\begin{bmatrix} s & -1.5 \\ 4 & s+4 \end{bmatrix}$$

Substituting all expressions into Eq. (1) yields

$$G(s) = \frac{3s + 5}{s^2 + 4s + 6}$$

3.5

Writing the differential equation we obtain

$$\frac{d^2x}{dt^2} + 2x^2 = 10 + \delta f(t) \tag{1}$$

Letting $x = x_o + \delta x$ and substituting into Eq. (1) yields

$$\frac{d^2(x_o + \delta x)}{dt^2} + 2(x_o + \delta x)^2 = 10 + \delta f(t) \tag{2}$$

Now, linearize x^2.

$$(x_o + \delta x)^2 - x_o^2 = \left.\frac{d(x^2)}{dx}\right|_{x_o} \delta x = 2x_o\delta x$$

from which

$$(x_o + \delta x)^2 = x_o^2 + 2x_o\delta x \tag{3}$$

Substituting Eq. (3) into Eq. (1) and performing the indicated differentiation gives us the linearized intermediate differential equation,

$$\frac{d^2\delta x}{dt^2} + 4x_o\delta x = -2x_o^2 + 10 + \delta f(t) \tag{4}$$

The force of the spring at equilibrium is 10 N. Thus, since $F = 2x^2, 10 = 2x_o^2$ from which

$$x_o = \sqrt{5}$$

Substituting this value of x_o into Eq. (4) gives us the final linearized differential equation.

$$\frac{d^2\delta x}{dt^2} + 4\sqrt{5}\,\delta x = \delta f(t)$$

Selecting the state variables,

$$x_1 = \delta x$$
$$x_2 = \dot{\delta} x$$

Writing the state and output equations

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \ddot{\delta} x = -4\sqrt{5}x_1 + \delta f(t)$$
$$y = x_1$$

Converting to vector-matrix form yields the final result as

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -4\sqrt{5} & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\delta f(t)$$
$$y = [1 \quad 0]\mathbf{x}$$

CHAPTER 4

4.1

For a step input

$$C(s) = \frac{10(s+4)(s+6)}{s(s+1)(s+7)(s+8)(s+10)} = \frac{A}{s} + \frac{B}{s+1} + \frac{C}{s+7} + \frac{D}{s+8} + \frac{E}{s+10}$$

Taking the inverse Laplace transform,

$$c(t) = A + Be^{-t} + Ce^{-7t} + De^{-8t} + Ee^{-10t}$$

4.2

Since $a = 50$, $T_c = \dfrac{1}{a} = \dfrac{1}{50} = 0.02\text{s}$; $T_s = \dfrac{4}{a} = \dfrac{4}{50} = 0.08\,\text{s}$; and $T_r = \dfrac{2.2}{a} = \dfrac{2.2}{50} = 0.044\,\text{s}$.

4.3

a. Since poles are at $-6 \pm j19.08$, $c(t) = A + Be^{-6t}\cos(19.08t + \phi)$.
b. Since poles are at -78.54 and -11.46, $c(t) = A + Be^{-78.54t} + Ce^{-11.4t}$.
c. Since poles are double on the real axis at -15 $c(t) = A + Be^{-15t} + Cte^{-15t}$.
d. Since poles are at $\pm j25$, $c(t) = A + B\cos(25t + \phi)$.

4.4

a. $\omega_n = \sqrt{400} = 20$ and $2\zeta\omega_n = 12$; $\therefore \zeta = 0.3$ and system is underdamped.
b. $\omega_n = \sqrt{900} = 30$ and $2\zeta\omega_n = 90$; $\therefore \zeta = 1.5$ and system is overdamped.
c. $\omega_n = \sqrt{225} = 15$ and $2\zeta\omega_n = 30$; $\therefore \zeta = 1$ and system is critically damped.
d. $\omega_n = \sqrt{625} = 25$ and $2\zeta\omega_n = 0$; $\therefore \zeta = 0$ and system is undamped.

4.5

$$\omega_n = \sqrt{361} = 19 \text{ and } 2\zeta\omega_n = 16; \quad \therefore \zeta = 0.421.$$

Now, $T_s = \dfrac{4}{\zeta\omega_n} = 0.5\,s$ and $T_p = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}} = 0.182$ s.

From Figure 4.16, $\omega_n T_r = 1.4998$. Therefore, $T_r = 0.079$ s.

Finally, $\%\text{os} = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} * 100 = 23.3\%$

4.6

a. The second-order approximation is valid, since the dominant poles have a real part of -2 and the higher-order pole is at -15, i.e. more than five-times further.

b. The second-order approximation is not valid, since the dominant poles have a real part of -1 and the higher-order pole is at -4, i.e. not more than five-times further.

4.7

a. Expanding $G(s)$ by partial fractions yields $G(s) = \dfrac{1}{s} + \dfrac{0.8942}{s+20} - \dfrac{1.5918}{s+10} - \dfrac{0.3023}{s+6.5}$. But -0.3023 is not an order of magnitude less than residues of second-order terms (term 2 and 3). Therefore, a second-order approximation is not valid.

b. Expanding $G(s)$ by partial fractions yields $G(s) = \dfrac{1}{s} + \dfrac{0.9782}{s+20} - \dfrac{1.9078}{s+10} - \dfrac{0.0704}{s+6.5}$. But 0.0704 is an order of magnitude less than residues of second-order terms (term 2 and 3). Therefore, a second-order approximation is valid.

4.8

See Figure 4.31 in the textbook for the Simulink block diagram and the output responses.

4.9

a. Since $s\mathbf{I} - \mathbf{A} = \begin{bmatrix} s & -2 \\ 3 & s+5 \end{bmatrix}$, $(s\mathbf{I} - \mathbf{A})^{-1} = \dfrac{1}{s^2+5s+6}\begin{bmatrix} s+5 & 2 \\ -3 & s \end{bmatrix}$. Also, $\mathbf{BU}(s) = \begin{bmatrix} 0 \\ 1/(s+1) \end{bmatrix}$.

The state vector is $\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}[\mathbf{x}(0) + \mathbf{BU}(s)] = \dfrac{1}{(s+1)(s+2)(s+3)} \times \begin{bmatrix} 2(s^2+7s+7) \\ s^2 - 4s - 6 \end{bmatrix}$. The output is $Y(s) = [1 \quad 3]\mathbf{X}(s) = \dfrac{5s^2+2s-4}{(s+1)(s+2)(s+3)} = -\dfrac{0.5}{s+1} - \dfrac{12}{s+2} + \dfrac{17.5}{s+3}$. Taking the inverse Laplace transform yields $y(t) = -0.5e^{-t} - 12e^{-2t} + 17.5e^{-3t}$.

b. The eigenvalues are given by the roots of $|s\mathbf{I} - \mathbf{A}| = s^2 + 5s + 6$, or -2 and -3.

4.10

a. Since $(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s & -2 \\ 2 & s+5 \end{bmatrix}$, $(s\mathbf{I} - \mathbf{A})^{-1} = \dfrac{1}{s^2+5s+4}\begin{bmatrix} s+5 & 2 \\ -2 & s \end{bmatrix}$. Taking the Laplace transform of each term, the state transition matrix is given by

$$\Phi(t) = \begin{bmatrix} \frac{4}{3}e^{-t} - \frac{1}{3}e^{-4t} & \frac{2}{3}e^{-t} - \frac{2}{3}e^{-4t} \\ -\frac{2}{3}e^{-t} + \frac{2}{3}e^{-4t} & -\frac{1}{3}e^{-t} + \frac{4}{3}e^{-4t} \end{bmatrix}.$$

b. Since $\Phi(t-\tau) = \begin{bmatrix} \frac{4}{3}e^{-(t-\tau)} - \frac{1}{3}e^{-4(t-\tau)} & \frac{2}{3}e^{-(t-\tau)} - \frac{2}{3}e^{-4(t-\tau)} \\ -\frac{2}{3}e^{-(t-\tau)} + \frac{2}{3}e^{-4(t-\tau)} & -\frac{1}{3}e^{-(t-\tau)} + \frac{4}{3}e^{-4(t-\tau)} \end{bmatrix}$ and

$$\mathbf{Bu}(\tau) = \begin{bmatrix} 0 \\ e^{-2\tau} \end{bmatrix}, \ \Phi(t-\tau)\mathbf{Bu}(\tau) = \begin{bmatrix} \frac{2}{3}e^{-\tau}e^{-t} - \frac{2}{3}e^{2\tau}e^{-4t} \\ -\frac{1}{3}e^{-\tau}e^{-t} + \frac{4}{3}e^{2\tau}e^{-4t} \end{bmatrix}.$$

Thus, $\mathbf{x}(t) = \Phi(t)\mathbf{x}(0) + \int_0^t \Phi(t-\tau)$

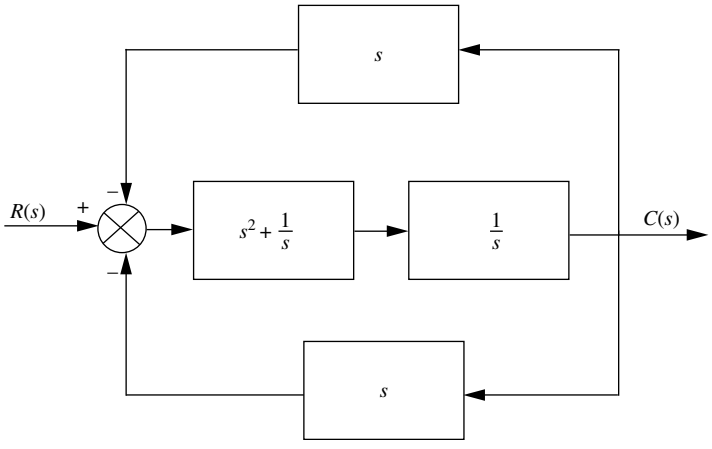
$$\mathbf{BU}(\tau)d\tau = \begin{bmatrix} \frac{10}{3}e^{-t} - e^{-2t} - \frac{4}{3}e^{-4t} \\ -\frac{5}{3}e^{-t} + e^{-2t} + \frac{8}{3}e^{-4t} \end{bmatrix}$$

c. $y(t) = [2 \quad 1]\mathbf{x} = 5e^{-t} - e^{-2t}$

CHAPTER 5

5.1

Combine the parallel blocks in the forward path. Then, push $\frac{1}{s}$ to the left past the pickoff point.



Combine the parallel feedback paths and get $2s$. Then, apply the feedback formula, simplify, and get, $T(s) = \dfrac{s^3+1}{2s^4+s^2+2s}$.

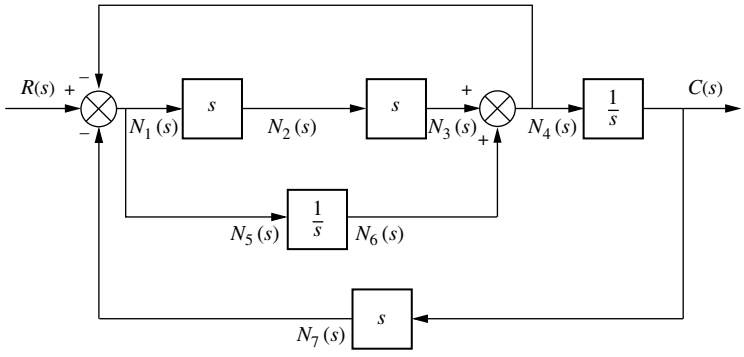
5.2

Find the closed-loop transfer function, $T(s) = \dfrac{G(s)}{1+G(s)H(s)} = \dfrac{16}{s^2+as+16}$, where and $G(s) = \dfrac{16}{s(s+a)}$ and $H(s) = 1$. Thus, $\omega_n = 4$ and $2\zeta\omega_n = a$, from which $\zeta = \dfrac{a}{8}$.

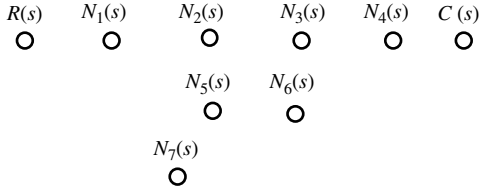
But, for 5% overshoot, $\zeta = \dfrac{-\ln\left(\frac{\%}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{\%}{100}\right)}} = 0.69$. Since, $\zeta = \dfrac{a}{8}$, $a = 5.52$.

5.3

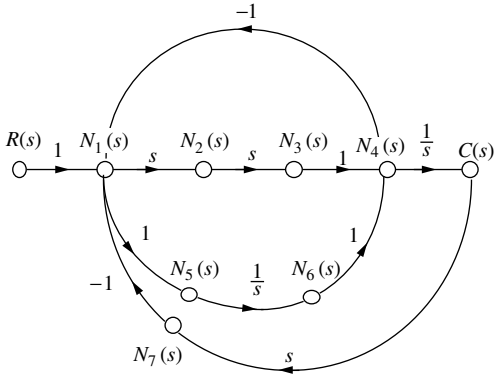
Label nodes.



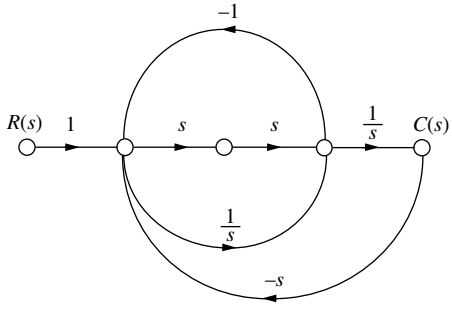
Draw nodes.



Connect nodes and label subsystems.



Eliminate unnecessary nodes.



5.4

Forward-path gains are $G_1G_2G_3$ and G_1G_3.
Loop gains are $-G_1G_2H_1$, $-G_2H_2$, and $-G_3H_3$.
Nontouching loops are $[-G_1G_2H_1][-G_3H_3] = G_1G_2G_3H_1H_3$ and $[-G_2H_2][-G_3H_3] = G_2G_3H_2H_3$.
Also, $\Delta = 1 + G_1G_2H_1 + G_2H_2 + G_3H_3 + G_1G_2G_3H_1H_3 + G_2G_3H_2H_3$.
Finally, $\Delta_1 = 1$ and $\Delta_2 = 1$.

Substituting these values into $T(s) = \dfrac{C(s)}{R(s)} = \dfrac{\sum_k T_k\Delta_k}{\Delta}$ yields

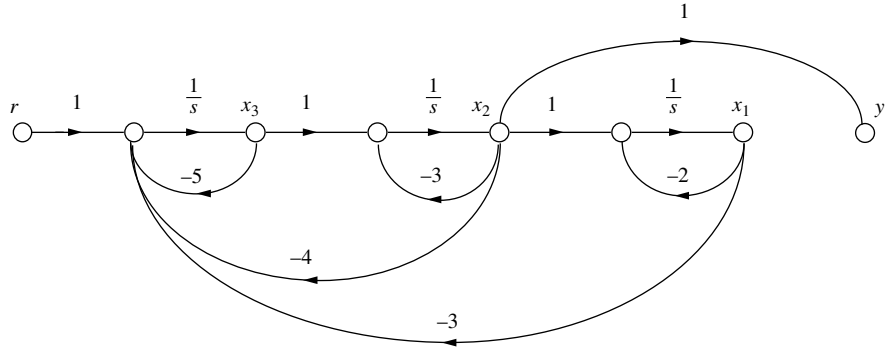
$$T(s) = \frac{G_1(s)G_3(s)[1 + G_2(s)]}{[1 + G_2(s)H_2(s) + G_1(s)G_2(s)H_1(s)][1 + G_3(s)H_3(s)]}$$

5.5

The state equations are,

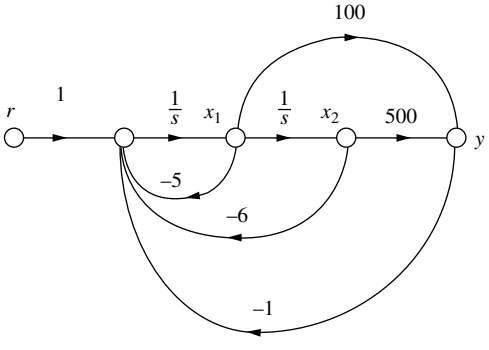
$$\begin{aligned}
\dot{x}_1 &= -2x_1 + x_2 \\
\dot{x}_2 &= -3x_2 + x_3 \\
\dot{x}_3 &= -3x_1 - 4x_2 - 5x_3 + r \\
y &= x_2
\end{aligned}$$

Drawing the signal-flow diagram from the state equations yields



5.6

From $G(s) = \dfrac{100(s + 5)}{s^2 + 5s + 6}$ we draw the signal-flow graph in controller canonical form and add the feedback.



Writing the state equations from the signal-flow diagram, we obtain

$$\mathbf{x} = \begin{bmatrix} -105 & -506 \\ 1 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r$$
$$y = [100 \quad 500]\mathbf{x}$$

5.7

From the transformation equations,

$$\mathbf{P}^{-1} = \begin{bmatrix} 3 & -2 \\ 1 & -4 \end{bmatrix}$$

Taking the inverse,

$$\mathbf{P} = \begin{bmatrix} 0.4 & -0.2 \\ 0.1 & -0.3 \end{bmatrix}$$

Now,

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{P} = \begin{bmatrix} 3 & -2 \\ 1 & -4 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ -4 & -6 \end{bmatrix}\begin{bmatrix} 0.4 & -0.2 \\ 0.1 & -0.3 \end{bmatrix} = \begin{bmatrix} 6.5 & -8.5 \\ 9.5 & -11.5 \end{bmatrix}$$
$$\mathbf{P}^{-1}\mathbf{B} = \begin{bmatrix} 3 & -2 \\ 1 & -4 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} -3 \\ -11 \end{bmatrix}$$
$$\mathbf{C}\mathbf{P} = [1 \quad 4]\begin{bmatrix} 0.4 & -0.2 \\ 0.1 & -0.3 \end{bmatrix} = [0.8 \quad -1.4]$$

Therefore,

$$\dot{\mathbf{z}} = \begin{bmatrix} 6.5 & -8.5 \\ 9.5 & -11.5 \end{bmatrix}\mathbf{z} + \begin{bmatrix} -3 \\ -11 \end{bmatrix} u$$
$$y = [0.8 \quad -1.4]\mathbf{z}$$

5.8

First find the eigenvalues.

$$|\lambda\mathbf{I} - \mathbf{A}| = \left|\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 1 & 3 \\ -4 & -6 \end{bmatrix}\right| = \begin{vmatrix} \lambda - 1 & -3 \\ 4 & \lambda + 6 \end{vmatrix} = \lambda^2 + 5\lambda + 6$$

From which the eigenvalues are -2 and -3.

Now use $\mathbf{A}x_i = \lambda x_i$ for each eigenvalue, λ.

Thus,

$$\begin{bmatrix} 1 & 3 \\ -4 & -6 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

For $\lambda = -2$,

$$3x_1 + 3x_2 = 0$$
$$-4x_1 - 4x_2 = 0$$

Thus $x_1 = -x_2$

For $\lambda = -3$

$$4x_1 + 3x_2 = 0$$
$$-4x_1 - 3x_2 = 0$$

Thus $x_1 = -x_2$ and $x_1 = -0.75x_2$; from which we let

$$\mathbf{P} = \begin{bmatrix} 0.707 & -0.6 \\ -0.707 & 0.8 \end{bmatrix}$$

Taking the inverse yields

$$\mathbf{P}^{-1} = \begin{bmatrix} 5.6577 & 4.2433 \\ 5 & 5 \end{bmatrix}$$

Hence,

$$\mathbf{D} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} = \begin{bmatrix} 5.6577 & 4.2433 \\ 5 & 5 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ -4 & -6 \end{bmatrix}\begin{bmatrix} 0.707 & -0.6 \\ -0.707 & 0.8 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}$$

$$\mathbf{P}^{-1}\mathbf{B} = \begin{bmatrix} 5.6577 & 4.2433 \\ 5 & 5 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 18.39 \\ 20 \end{bmatrix}$$

$$\mathbf{C}\mathbf{P} = \begin{bmatrix} 1 & 4 \end{bmatrix}\begin{bmatrix} 0.707 & -0.6 \\ -0.707 & 0.8 \end{bmatrix} = \begin{bmatrix} -2.121 & 2.6 \end{bmatrix}$$

Finally,

$$\dot{\mathbf{z}} = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 18.39 \\ 20 \end{bmatrix}u$$
$$\mathbf{y} = \begin{bmatrix} -2.121 & 2.6 \end{bmatrix}\mathbf{z}$$

CHAPTER 6

6.1

Make a Routh table.

s^7	3	6	7	2
s^6	9	4	8	6
s^5	4.666666667	4.333333333	0	0
s^4	−4.35714286	8	6	0
s^3	12.90163934	6.426229508	0	0
s^2	10.17026684	6	0	0
s^1	−1.18515742	0	0	0
s^0	6	0	0	0

Since there are four sign changes and no complete row of zeros, there are four right half-plane poles and three left half-plane poles.

6.2

Make a Routh table. We encounter a row of zeros on the s^3 row. The even polynomial is contained in the previous row as $-6s^4 + 0s^2 + 6$. Taking the derivative yields

$-24s^3 + 0s$. Replacing the row of zeros with the coefficients of the derivative yields the s^3 row. We also encounter a zero in the first column at the s^2 row. We replace the zero with ε and continue the table. The final result is shown now as

s^6	1	−6	−1	6	
s^5	1	0	−1	0	
s^4	6	0	6	0	
s^3	−24	0	0	0	ROZ
s^2	ε	6	0	0	
s^1	$144/\varepsilon$	0	0	0	
s^0	6	0	0	0	

There is one sign change below the even polynomial. Thus the even polynomial (4th order) has one right half-plane pole, one left half-plane pole, and 2 imaginary axis poles. From the top of the table down to the even polynomial yields one sign change. Thus, the rest of the polynomial has one right half-plane root, and one left half-plane root. The total for the system is two right half-plane poles, two left half-plane poles, and 2 imaginary poles.

6.3

Since $G(s) = \dfrac{K(s+20)}{s(s+2)(s+3)}$, $T(s) = \dfrac{G(s)}{1+G(s)} = \dfrac{K(s+20)}{s^3 + 5s^2 + (6+K)s + 20K}$

Form the Routh table.

s^3	1	$(6+K)$
s^2	5	$20K$
s^1	$\dfrac{30-15K}{5}$	
s^0	$20K$	

From the s^1 row, $K < 2$. From the s^0 row, $K > 0$. Thus, for stability, $0 < K < 2$.

6.4

First find

$$|s\mathbf{I} - \mathbf{A}| = \left| \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 2 & 1 & 1 \\ 1 & 7 & 1 \\ -3 & 4 & -5 \end{bmatrix} \right| = \begin{vmatrix} (s-2) & -1 & -1 \\ -1 & (s-7) & -1 \\ 3 & -4 & (s+5) \end{vmatrix}$$

$$= s^3 - 4s^2 - 33s + 51$$

Now form the Routh table.

s^3	1	−33
s^2	−4	51
s^1	−20.25	
s^0	51	

There are two sign changes. Thus, there are two rhp poles and one lhp pole.

CHAPTER 7

7.1

a. First check stability.

$$T(s) = \frac{G(s)}{1+G(s)} = \frac{10s^2+500s+6000}{s^3+70s^2+1375s+6000} = \frac{10(s+30)(s+20)}{(s+26.03)(s+37.89)(s+6.085)}$$

Poles are in the lhp. Therefore, the system is stable. Stability also could be checked via Routh-Hurwitz using the denominator of $T(s)$. Thus,

$$15u(t): \quad e_{step}(\infty) = \frac{15}{1+\lim\limits_{s\to 0} G(s)} = \frac{15}{1+\infty} = 0$$

$$15tu(t): \quad e_{ramp}(\infty) = \frac{15}{\lim\limits_{s\to 0} sG(s)} = \frac{15}{\frac{10^*20^*30}{25^*35}} = 2.1875$$

$$15t^2u(t): \quad e_{parabola}(\infty) = \frac{15}{\lim\limits_{s\to 0} s^2G(s)} = \frac{30}{0} = \infty, \text{ since } L\left[15t^2\right] = \frac{30}{s^3}$$

b. First check stability.

$$T(s) = \frac{G(s)}{1+G(s)} = \frac{10s^2+500s+6000}{s^5+110s^4+3875s^3+4.37e04s^2+500s+6000}$$
$$= \frac{10(s+30)(s+20)}{(s+50.01)(s+35)(s+25)(s^2-7.189e-04s+0.1372)}$$

From the second-order term in the denominator, we see that the system is unstable. Instability could also be determined using the Routh-Hurwitz criteria on the denominator of $T(s)$. Since the system is unstable, calculations about steady-state error cannot be made.

7.2

a. The system is stable, since

$$T(s) = \frac{G(s)}{1+G(s)} = \frac{1000(s+8)}{(s+9)(s+7)+1000(s+8)} = \frac{1000(s+8)}{s^2+1016s+8063}$$

and is of Type 0. Therefore,

$$K_p = \lim_{s\to 0} G(s) = \frac{1000^*8}{7^*9} = 127;\ K_v = \lim_{s\to 0} sG(s) = 0;$$

and $K_a = \lim\limits_{s\to 0} s^2G(s) = 0$

b.

$$e_{step}(\infty) = \frac{1}{1+\lim\limits_{s\to 0} G(s)} = \frac{1}{1+127} = 7.8e-03$$

$$e_{ramp}(\infty) = \frac{1}{\lim\limits_{s\to 0} sG(s)} = \frac{1}{0} = \infty$$

$$e_{parabola}(\infty) = \frac{1}{\lim\limits_{s\to 0} s^2G(s)} = \frac{1}{0} = \infty$$

7.3

System is stable for positive K. System is Type 0. Therefore, for a step input $e_{step}(\infty) = \dfrac{1}{1+K_p} = 0.1$. Solving for K_p yields $K_p = 9 = \lim\limits_{s\to 0} G(s) = \dfrac{12K}{14^*18}$; from which we obtain $K = 189$.

7.4

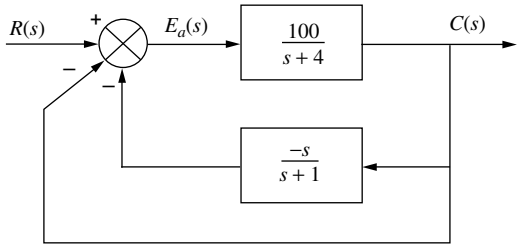
System is stable. Since $G_1(s) = 1000$, and $G_2(s) = \dfrac{(s+2)}{(s+4)}$,

$$e_D(\infty) = -\frac{1}{\lim\limits_{s\to 0}\dfrac{1}{G_2(s)} + \lim\limits_{s\to 0} G_1(s)} = \frac{1}{2+1000} = -9.98e-04$$

7.5

System is stable. Create a unity-feedback system, where $H_e(s) = \dfrac{1}{s+1} - 1 = \dfrac{-s}{s+1}$

The system is as follows:



Thus,

$$G_e(s) = \frac{G(s)}{1+G(S)H_e(s)} = \frac{\dfrac{100}{(s+4)}}{1-\dfrac{100s}{(s+1)(s+4)}} = \frac{100(s+1)}{S^2-95s+4}$$

Hence, the system is Type 0. Evaluating K_p yields

$$K_p = \frac{100}{4} = 25$$

The steady-state error is given by

$$e_{step}(\infty) = \frac{1}{1+K_p} = \frac{1}{1+25} = 3.846e-02$$

7.6

Since $G(s) = \dfrac{K(s+7)}{s^2+2s+10}$, $e(\infty) = \dfrac{1}{1+K_p} = \dfrac{1}{1+\dfrac{7K}{10}} = \dfrac{10}{10+7K}$.

Calculating the sensitivity, we get

$$S_{e:K} = \frac{K}{e}\frac{\partial e}{\partial K} = \frac{K}{\left(\dfrac{10}{10+7K}\right)}\frac{(-10)7}{(10+7K)^2} = -\frac{7K}{10+7K}$$

7.7

Given

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -3 & -6 \end{bmatrix};\ \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix};\ \mathbf{C} = [1 \quad 1];\ \mathrm{R(s)} = \frac{1}{\mathrm{s}}.$$

Using the final value theorem,

$$e_{step}(\infty) = \lim_{s\to 0} sR(s)\left[1 - \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\right] = \lim_{s\to 0}\left[1 - [1 \quad 1]\begin{bmatrix} s & -1 \\ 3 & s+6 \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right]$$

$$= \lim_{s\to 0}\left[1 - [1 \quad 1]\frac{\begin{bmatrix} s+6 & 1 \\ -3s & s \end{bmatrix}}{s^2+6s+3}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right] = \lim_{s\to 0}\frac{s^2+5s+2}{s^2+6s+3} = \frac{2}{3}$$

Using input substitution,

$$_{step}(\infty) = 1 + \mathbf{C}\mathbf{A}^{-1}\mathbf{B} = 1 - [1 \quad 1]\begin{bmatrix} 0 & 1 \\ -3 & -6 \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= 1 + [1 \quad 1]\frac{\begin{bmatrix} -6 & -1 \\ 3 & 0 \end{bmatrix}}{3}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 + [1 \quad 1]\begin{bmatrix} -\frac{1}{3} \\ 0 \end{bmatrix} = \frac{2}{3}$$

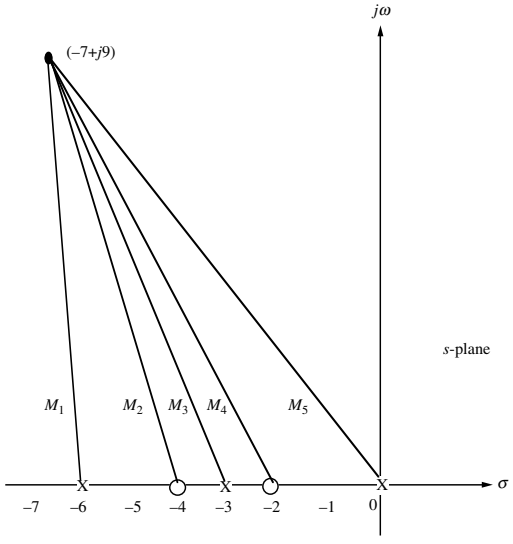
CHAPTER 8

8.1

a. $$F(-7+j9) = \frac{(-7+j9+2)(-7+j9+4)0.0339}{(-7+j9)(-7+j9+3)(-7+j9+6)} = \frac{(-5+j9)(-3+j9)}{(-7+j9)(-4+j9)(-1+j9)}$$

$$= \frac{(-66-j72)}{(944-j378)} = -0.0339 - j0.0899 = 0.096 < -110.7^\circ$$

b. The arrangement of vectors is shown as follows:

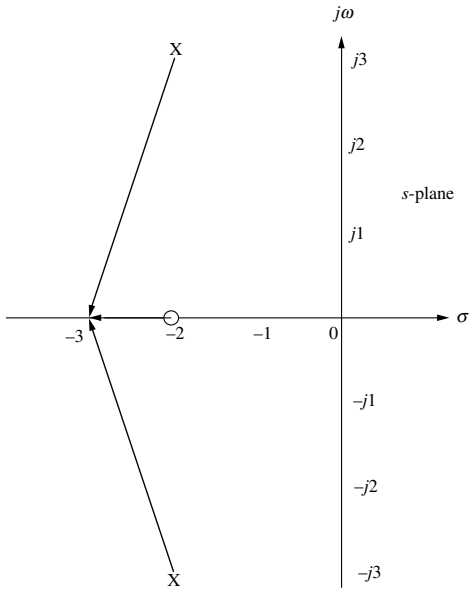


From the diagram,

$$F(-7+j9) = \frac{M_2M_4}{M_1M_3M_5} = \frac{(-3+j9)(-5+j9)}{(-1+j9)(-4+j9)(-7+j9)} = \frac{(-66-j72)}{(944-j378)}$$
$$= -0.0339 - j0.0899 = 0.096 <; -110.7^\circ$$

8.2

a. First draw the vectors.



From the diagram,

$$\sum \text{ angles} = 180^\circ - \tan^{-1}\left(\frac{-3}{-1}\right) - \tan^{-1}\left(\frac{-3}{1}\right) = 180^\circ - 108.43^\circ + 108.43^\circ = 180^\circ.$$

b. Since the angle is 180°, the point is on the root locus.

c. $K = \dfrac{\Pi \text{ pole lengths}}{\Pi \text{ zero lengths}} = \dfrac{\left(\sqrt{1^2+3^2}\right)\left(\sqrt{1^2+3^2}\right)}{1} = 10$

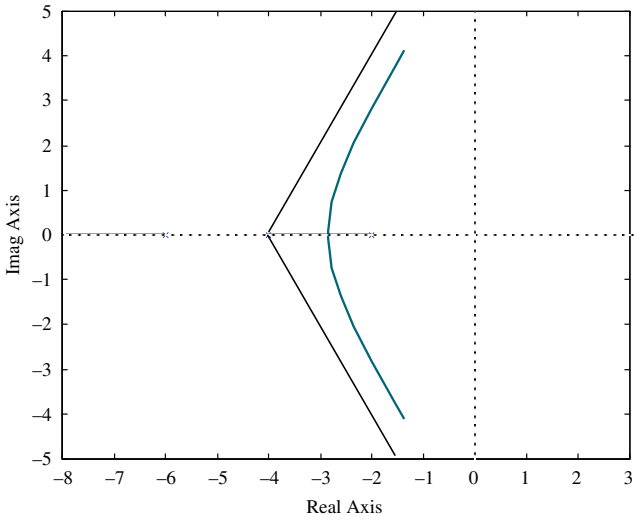
8.3

First, find the asymptotes.

$$\sigma_a = \frac{\sum \text{poles} - \sum \text{zeros}}{\#\text{ poles} - \#\text{ zeros}} = \frac{(-2-4-6)-(0)}{3-0} = -4$$

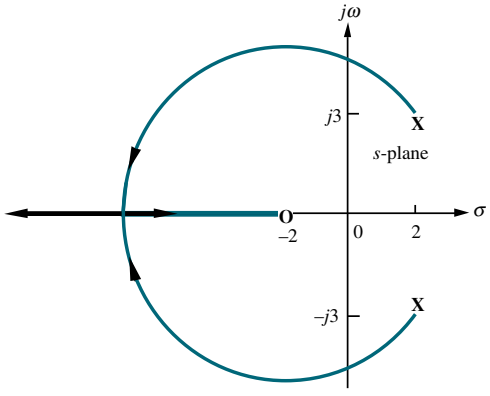
$$\theta_a = \frac{(2k+1)\pi}{3} = \frac{\pi}{3},\ \pi,\ \frac{5\pi}{3}$$

Next draw root locus following the rules for sketching.



8.4

a.



b. Using the Routh-Hurwitz criteria, we first find the closed-loop transfer function.

$$T(s) = \frac{G(s)}{1 + G(s)} = \frac{K(s+2)}{s^2 + (K-4)s + (2K+13)}$$

Using the denominator of T(s), make a Routh table.

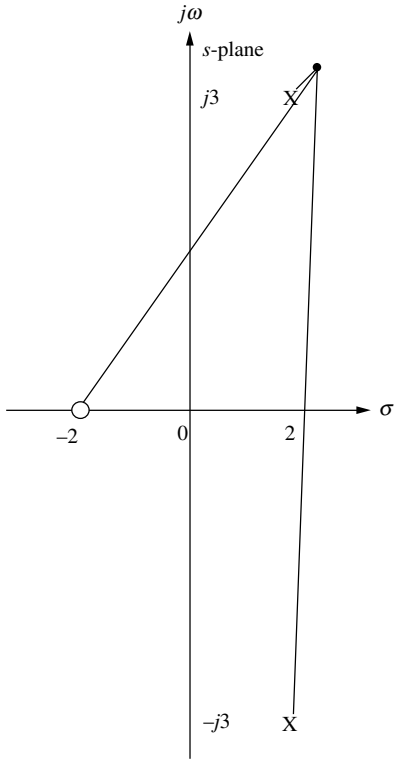
s^2	1	$2K + 13$
s^1	$K - 40$	0
s^0	$2K + 13$	0

We get a row of zeros for $K = 4$. From the s^2 row with $K = 4$, $s^2 + 21 = 0$. From which we evaluate the imaginary axis crossing at $\sqrt{21}$.

c. From part (b), $K = 4$.

d. Searching for the minimum gain to the left of -2 on the real axis yields -7 at a gain of 18. Thus the break-in point is at -7.

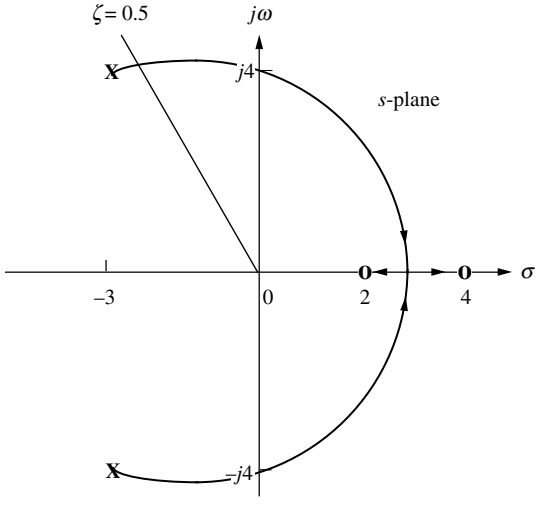
e. First, draw vectors to a point ε close to the complex pole.



At the point ε close to the complex pole, the angles must add up to zero. Hence, angle from zero – angle from pole in 4th quadrant – angle from pole in 1st quadrant = 180°, or $\tan^{-1}\left(\frac{3}{4}\right) - 90^\circ - \theta = 180^\circ$. Solving for the angle of departure, $\theta = -233.1$.

8.5

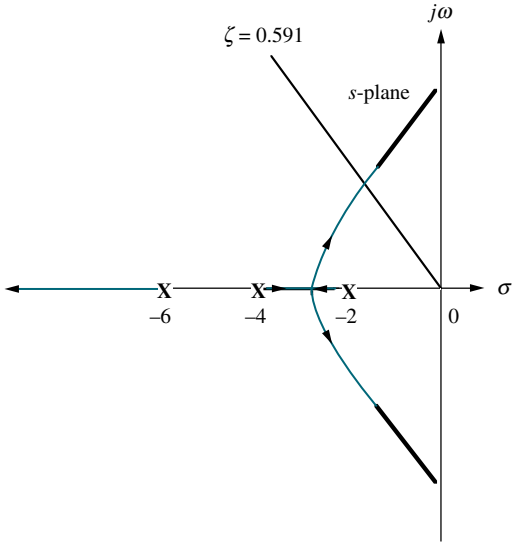
a.



b. Search along the imaginary axis and find the 180° point at $s = \pm j4.06$.
c. For the result in part (b), $K = 1$.
d. Searching between 2 and 4 on the real axis for the minimum gain yields the break-in at $s = 2.89$.
e. Searching along $\zeta = 0.5$ for the 180° point we find $s = -2.42 + j4.18$.
f. For the result in part (e), $K = 0.108$.
g. Using the result from part (c) and the root locus, $K < 1$.

8.6

a.



b. Searching along the $\zeta = 0.591$ (10% overshoot) line for the 180° point yields $-2.028 + j2.768$ with $K = 45.55$.
c. $T_s = \dfrac{4}{|\text{Re}|} = \dfrac{4}{2.028} = 1.97\,s$; $T_p = \dfrac{\pi}{|\text{Im}|} = \dfrac{\pi}{2.768} = 1.13\,s$; $\omega_n T_r = 1.8346$ from the rise-time chart and graph in Chapter 4. Since ω_n is the radial distance to the pole, $\omega_n = \sqrt{2.028^2 + 2.768^2} = 3.431$. Thus, $T_r = 0.53\,s$; since the system is Type 0, $K_p = \dfrac{K}{2^*4^*6} = \dfrac{45.55}{48} = 0.949$. Thus,

$$e_{step}(\infty) = \frac{1}{1 + K_p} = 0.51.$$

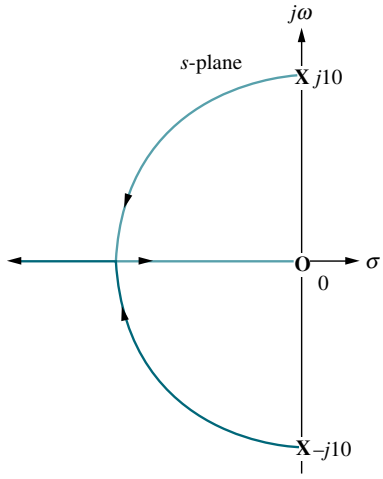
d. Searching the real axis to the left of -6 for the point whose gain is 45.55, we find -7.94. Comparing this value to the real part of the dominant pole, -2.028, we find that it is not five times further. The second-order approximation is not valid.

8.7

Find the closed-loop transfer function and put it the form that yields p_i as the root locus variable. Thus,

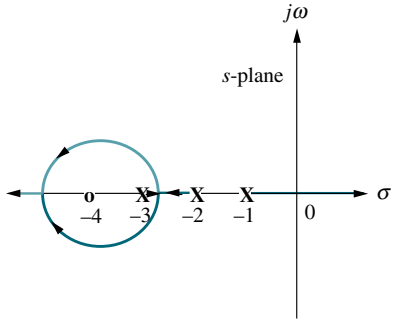
$$T(s) = \frac{G(s)}{1 + G(s)} = \frac{100}{s^2 + p_i s + 100} = \frac{100}{(s^2 + 100) + p_i s} = \frac{\dfrac{100}{s^2 + 100}}{1 + \dfrac{p_i s}{s^2 + 100}}$$

Hence, $KG(s)H(s) = \dfrac{p_i s}{s^2 + 100}$. The following shows the root locus.



8.8

Following the rules for plotting the root locus of positive-feedback systems, we obtain the following root locus:



8.9

The closed-loop transfer function is $T(s) = \dfrac{K(s+1)}{s^2 + (K+2)s + K}$. Differentiating the denominator with respect to K yields

$$2s\frac{\partial s}{\partial K} + (K+2)\frac{\partial s}{\partial K} + (s+1) = (2s + K + 2)\frac{\partial s}{\partial K} + (s+1) = 0$$

Solving for $\dfrac{\partial s}{\partial K}$, we get $\dfrac{\partial s}{\partial K} = \dfrac{-(s+1)}{(2s+K+2)}$. Thus, $S_{s:K} = \dfrac{K}{s}\dfrac{\partial s}{\partial K} = \dfrac{-K(s+1)}{s(2s+K+2)}$.

Substituting $K = 20$ yields $S_{s:K} = \dfrac{-10(s+1)}{s(s+11)}$.

Now find the closed-loop poles when $K = 20$. From the denominator of $T(s)$, $s_{1,2} = -21.05, \ -0.95$, when $K = 20$.

For the pole at -21.05,

$$\Delta s = s(S_{s:K})\frac{\Delta K}{K} = -12.05\left(\frac{-10(-21.05+1)}{-21.05(-21.05+11)}\right)0.05 = -0.9975.$$

For the pole at -0.95,

$$\Delta s = s(S_{s:K})\frac{\Delta K}{K} = 0.95\left(\frac{-10(-0.95+1)}{-0.95(-0.95+11)}\right)0.05 = -0.0025.$$

CHAPTER 9

9.1

a. Searching along the 15% overshoot line, we find the point on the root locus at $-3.5 + j5.8$ at a gain of $K = 45.84$. Thus, for the uncompensated system, $K_v = \lim_{s\to 0} sG(s) = K/7 = 45.84/7 = 6.55$.

Hence, $e_{ramp\_uncompensated}(\infty) = 1/K_v = 0.1527$.

b. Compensator zero should be $20x$ further to the left than the compensator pole. Arbitrarily select $G_c(s) = \dfrac{(s+0.2)}{(s+0.01)}$.

c. Insert compensator and search along the 15% overshoot line and find the root locus at $-3.4 + j5.63$ with a gain, $K = 44.64$. Thus, for the compensated system, $K_v = \dfrac{44.64(0.2)}{(7)(0.01)} = 127.5$ and $e_{ramp\_compensated}(\infty) = \dfrac{1}{K_v} = 0.0078$.

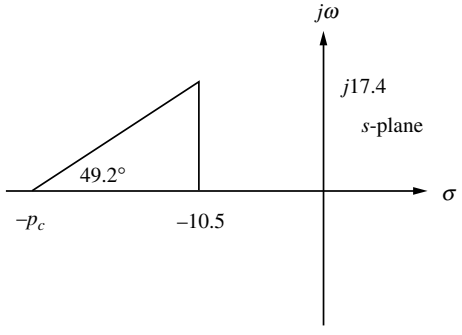
d. $\dfrac{e_{ramp\_uncompensated}}{e_{ramp\_compensated}} = \dfrac{0.1527}{0.0078} = 19.58$

9.2

a. Searching along the 15% overshoot line, we find the point on the root locus at $-3.5 + j5.8$ at a gain of $K = 45.84$. Thus, for the uncompensated system,

$$T_s = \frac{4}{|Re|} = \frac{4}{3.5} = 1.143\,s.$$

b. The real part of the design point must be three times larger than the uncompensated pole's real part. Thus the design point is $3(-3.5) + j\,3(5.8) = -10.5 + j\,17.4$. The angular contribution of the plant's poles and compensator zero at the design point is 130.8°. Thus, the compensator pole must contribute $180° - 130.8° = 49.2°$. Using the following diagram,



we find $\dfrac{17.4}{P_c - 10.5} = \tan 49.2°$, from which, $p_c = 25.52$. Adding this pole, we find the gain at the design point to be $K = 476.3$. A higher-order closed-loop pole is found to be at -11.54. This pole may not be close enough to the closed-loop zero at -10. Thus, we should simulate the system to be sure the design requirements have been met.

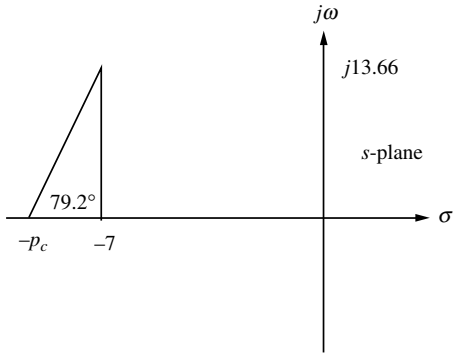
9.3

a. Searching along the 20% overshoot line, we find the point on the root locus at $-3.5 + 6.83$ at a gain of $K = 58.9$. Thus, for the uncompensated system,

$$T_s = \frac{4}{|\text{Re}|} = \frac{4}{3.5} = 1.143\,s.$$

b. For the uncompensated system, $K_v = \lim_{s\to 0} sG(s) = K/7 = 58.9/7 = 8.41$. Hence, $e_{ramp\_uncompensated}(\infty) = 1/K_v = 0.1189$.

c. In order to decrease the settling time by a factor of 2, the design point is twice the uncompensated value, or $-7 + j\,13.66$. Adding the angles from the plant's poles and the compensator's zero at -3 to the design point, we obtain $-100.8°$. Thus, the compensator pole must contribute $180° - 100.8° = 79.2°$. Using the following diagram,



we find $\dfrac{13.66}{P_c - 7} = \tan 79.2°$, from which, $p_c = 9.61$. Adding this pole, we find the gain at the design point to be $K = 204.9$.

Evaluating K_v for the lead-compensated system:

$$K_v = \lim_{s\to 0} sG(s)G_{lead} = K(3)/[(7)(9.61)] = (204.9)(3)/[(7)(9.61)] = 9.138.$$

K_v for the uncompensated system was 8.41. For a $10x$ improvement in steady-state error, K_v must be $(8.41)(10) = 84.1$. Since lead compensation gave us $K_v = 9.138$, we need an improvement of $84.1/9.138 = 9.2$. Thus, the lag compensator zero should be $9.2x$ further to the left than the compensator pole. Arbitrarily select $G_c(s) = \dfrac{(s + 0.092)}{(s + 0.01)}$.

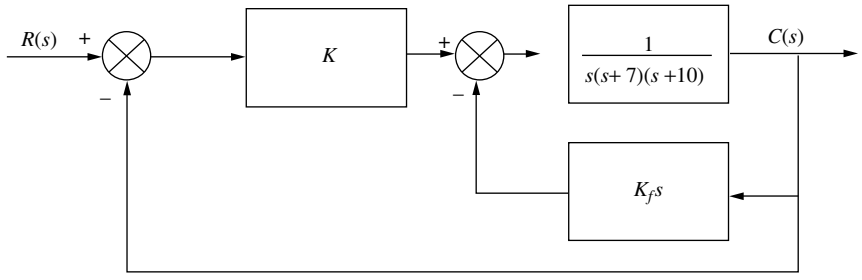
Using all plant and compensator poles, we find the gain at the design point to be $K = 205.4$. Summarizing the forward path with plant, compensator, and gain yields

$$G_e(s) = \frac{205.4(s + 3)(s + 0.092)}{s(s + 7)(9.61)(s + 0.01)}.$$

Higher-order poles are found at -0.928 and -2.6. It would be advisable to simulate the system to see if there is indeed pole-zero cancellation.

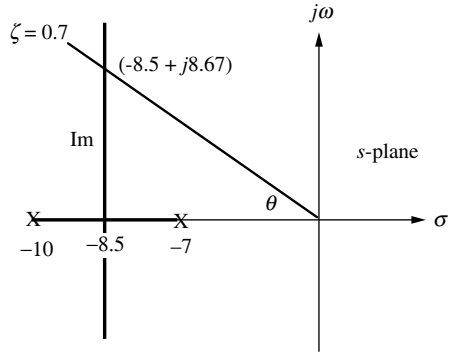
9.4

The configuration for the system is shown in the figure below.



Minor-Loop Design:

For the minor loop, $G(s)H(s) = \dfrac{K_f}{(s+7)(s+10)}$. Using the following diagram, we find that the minor-loop root locus intersects the 0.7 damping ratio line at $-8.5 + j\,8.67$. The imaginary part was found as follows: $\theta = \cos^{-1}\zeta = 45.57^\circ$. Hence, $\dfrac{\text{Im}}{8.5} = \tan 45.57^\circ$, from which Im $= 8.67$.



The gain, K_f, is found from the vector lengths as

$$K_f = \sqrt{1.5^2 + 8.67^2}\sqrt{1.5^2 + 8.67^2} = 77.42$$

Major-Loop Design:

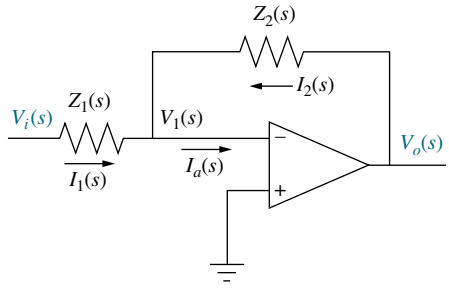
Using the closed-loop poles of the minor loop, we have an equivalent forward-path transfer function of

$$G_e(s) = \frac{K}{s(s + 8.5 + j8.67)(s + 8.5 - j8.67)} = \frac{K}{s(s^2 + 17s + 147.4)}.$$

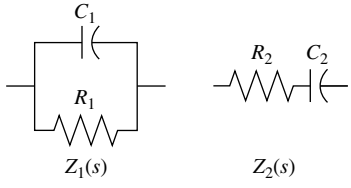
Using the three poles of $G_e(s)$ as open-loop poles to plot a root locus, we search along $\zeta = 0.5$ and find that the root locus intersects this damping ratio line at $-4.34 + j7.51$ at a gain, $K = 626.3$.

9.5

a. An active PID controller must be used. We use the circuit shown in the following figure:



where the impedances are shown below as follows:



Matching the given transfer function with the transfer function of the PID controller yields

$$G_c(s) = \frac{(s+0.1)(s+5)}{s} = \frac{s^2+5.1s+0.5}{s} = s + 5.1 + \frac{0.5}{8}$$

$$= -\left[\left(\frac{R_2}{R_1}+\frac{C_1}{C_2}\right) + R_2C_1s + \frac{\dfrac{1}{R_1C_2}}{s}\right]$$

Equating coefficients

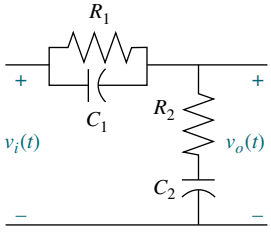
$$\frac{1}{R_1C_2} = 0.5 \tag{1}$$

$$R_2C_1 = 1 \tag{2}$$

$$\left(\frac{R_2}{R_1}+\frac{C_1}{C_2}\right) = 5.1 \tag{3}$$

In Eq. (2) we arbitrarily let $C_1 = 10^{-5}$. Thus, $R_2 = 10^5$. Using these values along with Eqs. (1) and (3) we find $C_2 = 100\mu F$ and $R_1 = 20\,\text{k}\Omega$.

b. The lag-lead compensator can be implemented with the following passive network, since the ratio of the lead pole-to-zero is the inverse of the ratio of the lag pole-to-zero:



Matching the given transfer function with the transfer function of the passive lag-lead compensator yields

$$G_c(s) = \frac{(s+0.1)(s+2)}{(s+0.01)(s+20)} = \frac{(s+0.1)(s+2)}{s^2+20.01s+0.2}$$

$$= \frac{\left(s+\frac{1}{R_1C_1}\right)\left(s+\frac{1}{R_2C_2}\right)}{s^2+\left(\frac{1}{R_1C_1}+\frac{1}{R_2C_2}+\frac{1}{R_2C_1}\right)s+\frac{1}{R_1R_2C_1C_2}}$$

Equating coefficients

$$\frac{1}{R_1C_1} = 0.1 \tag{1}$$

$$\frac{1}{R_2C_2} = 0.1 \tag{2}$$

$$\left(\frac{1}{R_1C_1}+\frac{1}{R_2C_2}+\frac{1}{R_2C_1}\right) = 20.01 \tag{3}$$

Substituting Eqs. (1) and (2) in Eq. (3) yields

$$\frac{1}{R_2C_1} = 17.91 \tag{4}$$

Arbitrarily letting $C_1 = 100\,\mu F$ in Eq. (1) yields $R_1 = 100\,k\Omega$.
Substituting $C_1 = 100\,\mu F$ into Eq. (4) yields $R_2 = 558\,k\Omega$.
Substituting $R_2 = 558\,k\Omega$ into Eq. (2) yields $C_2 = 900\,\mu F$.

CHAPTER 10

10.1

a.

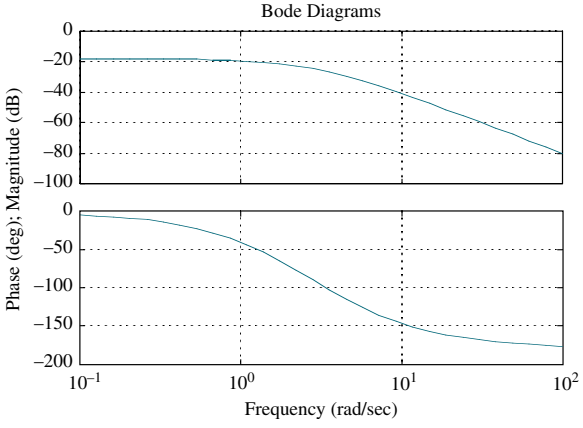
$$G(s) = \frac{1}{(s+2)(s+4)};\ G(j\omega) = \frac{1}{(8+\omega^2)+j6\omega}$$

$$M(\omega) = \sqrt{(8-\omega^2)^2+(6\omega)^2}$$

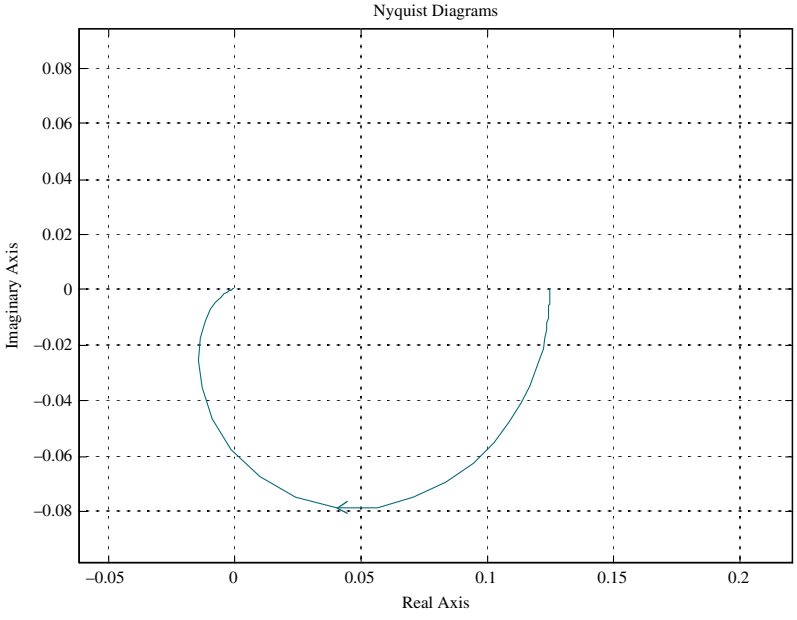
For $\omega < \sqrt{8}$, $\phi(\omega) = -\tan^{-1}\left(\frac{6\omega}{8-\omega^2}\right)$.

For $\omega < \sqrt{8}$, $\phi(\omega) = -\left(\pi + \tan^{-1}\left[\frac{6\omega}{8-\omega^2}\right]\right)$.

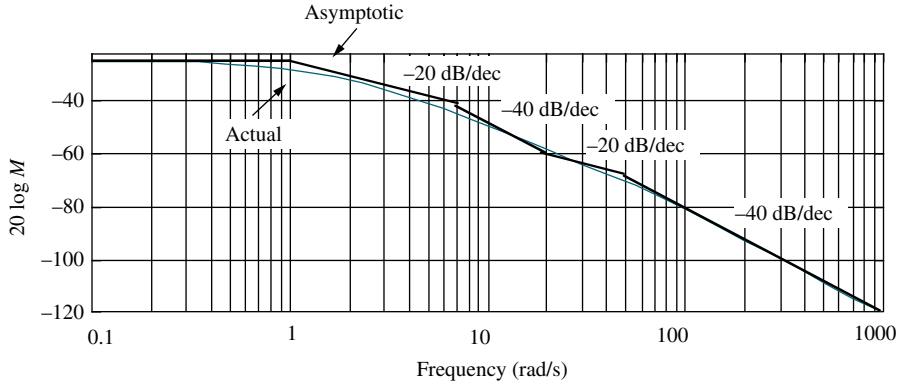
b.

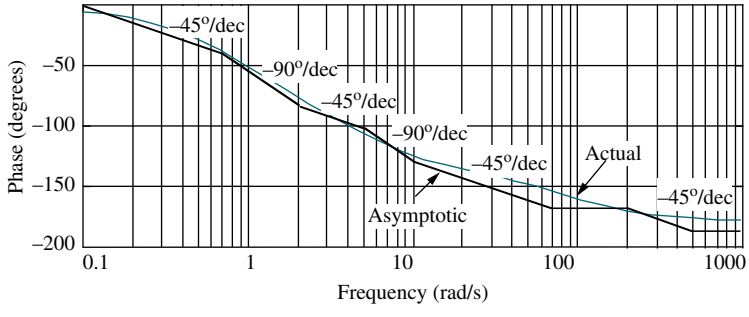


c.


Nyquist Diagrams
Imaginary Axis
Real Axis
0.08
0.06
0.04
0.02
0
–0.02
–0.04
–0.06
–0.08
–0.05
0
0.05
0.1
0.15
0.2

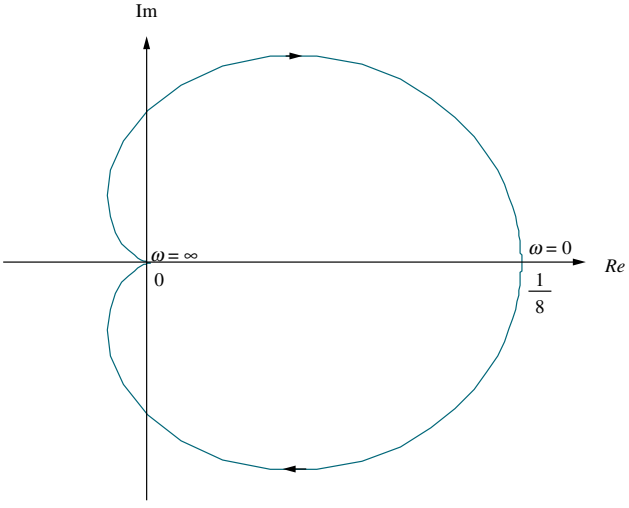
10.2


Asymptotic
–20 dB/dec
–40 dB/dec
–20 dB/dec
–40 dB/dec
Actual
20 log M
–40
–60
–80
–100
–120
0.1
1
10
100
1000
Frequency (rad/s)


–45°/dec
–90°/dec
–45°/dec
–90°/dec
–45°/dec
Actual
–45°/dec
Asymptotic
Phase (degrees)
–50
–100
–150
–200
0.1
1
10
100
1000
Frequency (rad/s)

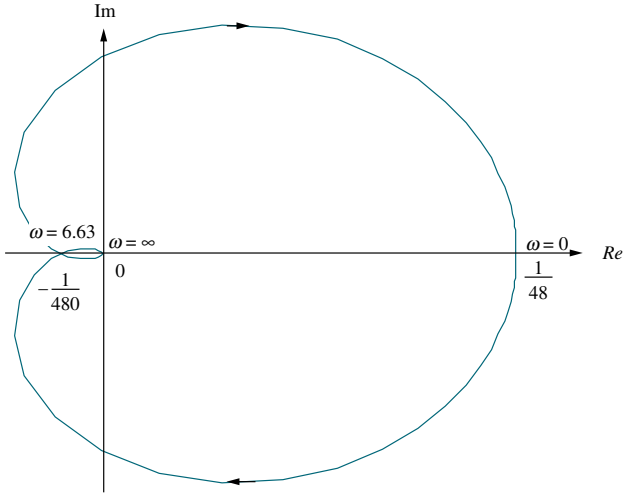
10.3

The frequency response is 1/8 at an angle of zero degrees at $\omega = 0$. Each pole rotates 90° in going from $\omega = 0$ to $\omega = \infty$. Thus, the resultant rotates -180° while its magnitude goes to zero. The result is shown below.



10.4

a. The frequency response is 1/48 at an angle of zero degrees at $\omega = 0$. Each pole rotates 90° in going from $\omega = 0$ to $\omega = \infty$. Thus, the resultant rotates -270° while its magnitude goes to zero. The result is shown below.



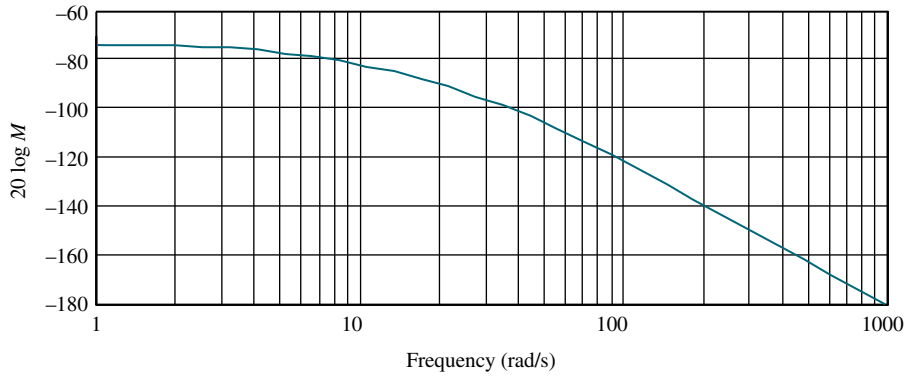
b. Substituting $j\omega$ into $G(s) = \dfrac{1}{(s+2)(s+4)(s+6)} = \dfrac{1}{s^3 + 12s^2 + 44s + 48}$ and simplifying, we obtain $G(j\omega) = \dfrac{(48 - 12\omega^2) - j(44\omega - \omega^3)}{\omega^6 + 56\omega^4 + 784\omega^2 + 2304}$. The Nyquist diagram crosses the real axis when the imaginary part of $G(j\omega)$ is zero. Thus, the Nyquist diagram crosses the real axis at $\omega^2 = 44$, or $\omega = \sqrt{44} = 6.63$ rad/s. At this frequency $G(j\omega) = -\dfrac{1}{480}$. Thus, the system is stable for $K < 480$.

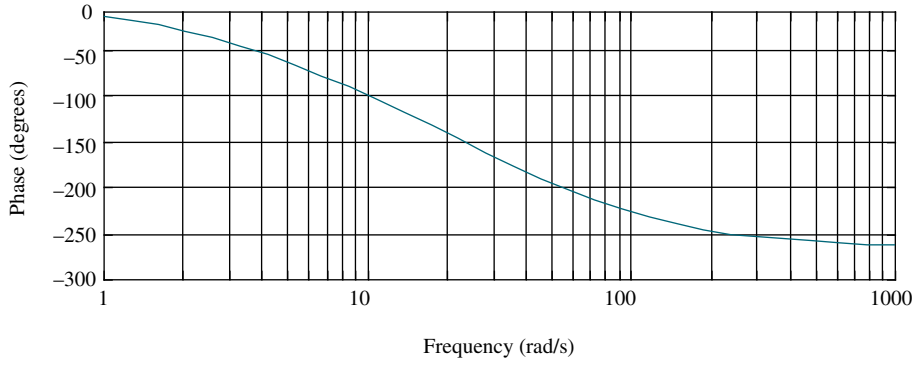
10.5

If $K = 100$, the Nyquist diagram will intersect the real axis at $-100/480$. Thus, $G_M = 20\log\dfrac{480}{100} = 13.62$ dB. From Skill-Assessment Exercise Solution 10.4, the 180° frequency is 6.63 rad/s.

10.6

a.





b. The phase angle is 180° at a frequency of 36.74 rad/s. At this frequency the gain is −99.67 dB. Therefore, $20\log K = 99.67$, or $K = 96,270$. We conclude that the system is stable for $K < 96,270$.

c. For $K = 10,000$, the magnitude plot is moved up $20\log 10,000 = 80$ dB. Therefore, the gain margin is $99.67 - 80 = 19.67$ dB. The 180° frequency is 36.7 rad/s. The gain curve crosses 0 dB at $\omega = 7.74$ rad/s, where the phase is 87.1°. We calculate the phase margin to be $180° - 87.1° = 92.9°$.

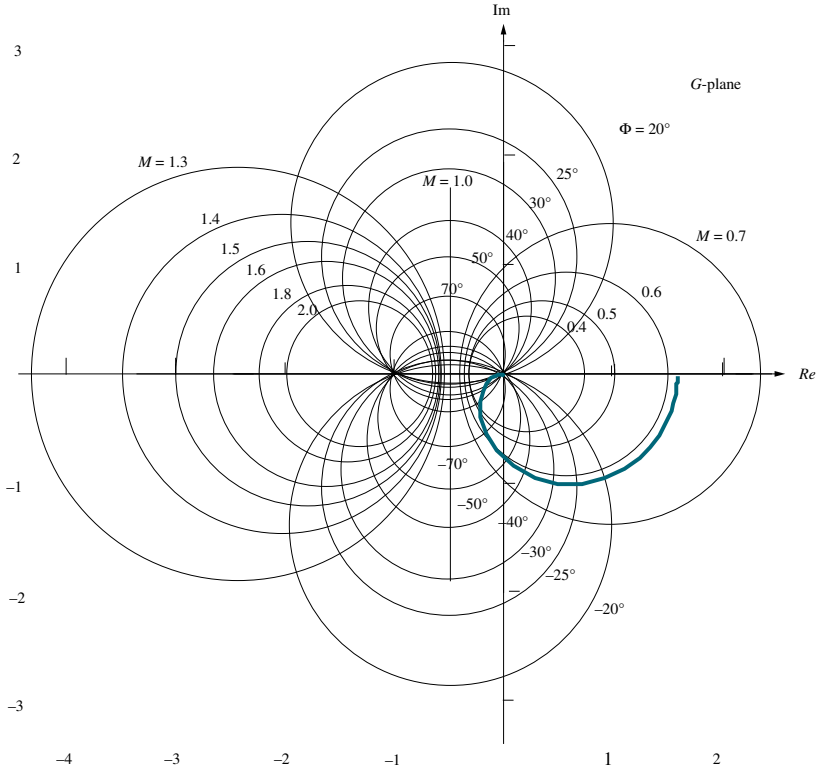
10.7

Using $\zeta = \dfrac{-\ln(\%/100)}{\sqrt{\pi^2 + \ln^2(\%100)}}$, we find $\zeta = 0.456$, which corresponds to 20% overshoot. Using $T_s = 2$, $\omega_{BW} = \dfrac{4}{T_s\zeta}\sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} = 5.79$ rad/s.

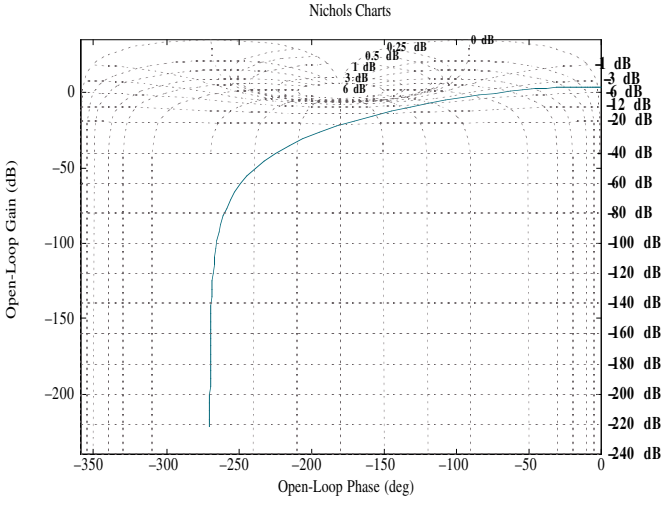
10.8

For both parts find that $G(j\omega) = \dfrac{160}{27} * \dfrac{(6750000 - 101250\omega^2) + j1350(\omega^2 - 1350)\omega}{\omega^6 + 2925\omega^4 + 1072500\omega^2 + 25000000}$.
For a range of values for ω, superimpose $G(j\omega)$ on the **a.** M and N circles, and on the **b.** Nichols chart.

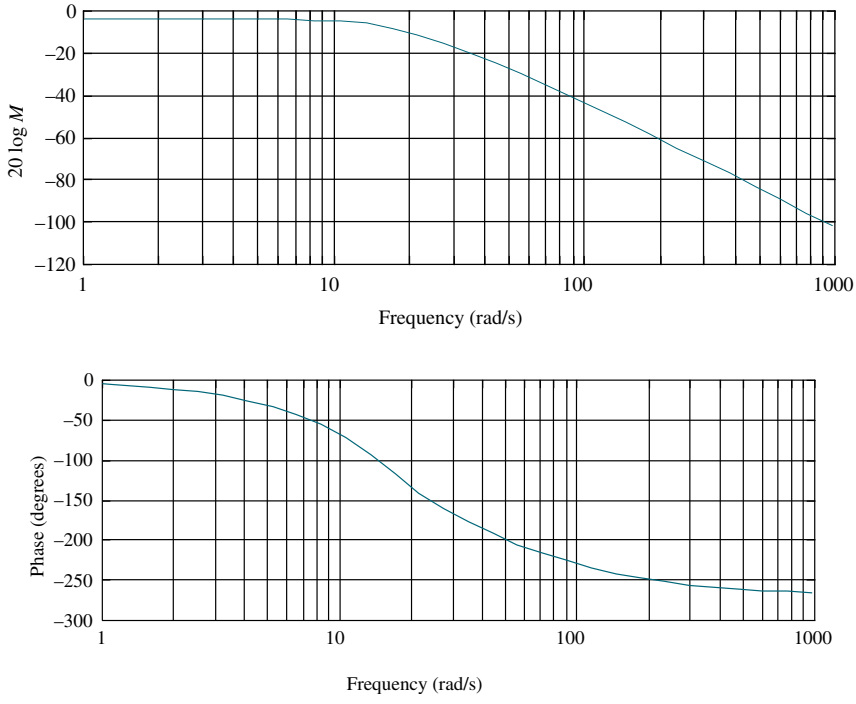
a.



b.

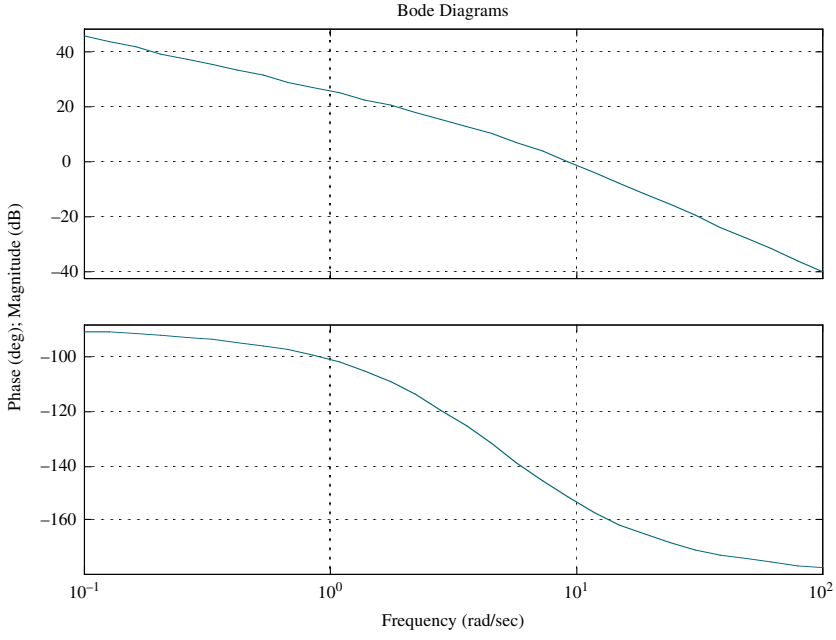


Plotting the closed-loop frequency response from **a.** or **b.** yields the following plot:



10.9

The open-loop frequency response is shown in the following figure:



The open-loop frequency response is -7 at $\omega = 14.5$ rad/s. Thus, the estimated bandwidth is $\omega_{WB} = 14.5$ rad/s. The open-loop frequency response plot goes through zero dB at a frequency of 9.4 rad/s, where the phase is 151.98°. Hence, the phase margin is $180° - 151.98° = 28.02°$. This phase margin corresponds to

$$\zeta = 0.25. \text{ Therefore, } \%OS = e^{-\left(\zeta\pi/\sqrt{1-\zeta^2}\right)} \times 100 = 44.4\%$$

$$T_s = \frac{4}{\omega_{BW}\zeta}\sqrt{(1-2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} = 1.64 \text{ s and}$$

$$T_p = \frac{\pi}{\omega_{BW}\sqrt{1-\zeta^2}}\sqrt{(1-2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} = 0.33 \text{ s}$$

10.10

The initial slope is 40 dB/dec. Therefore, the system is Type 2. The initial slope intersects 0 dB at $\omega = 9.5$ rad/s. Thus, $K_a = 9.5^2 = 90.25$ and $K_p = K_v = \infty$.

10.11

a. Without delay, $G(j\omega) = \dfrac{10}{j\omega(j\omega+1)} = \dfrac{10}{\omega(-\omega+j)}$, from which the zero dB frequency is found as follows: $M = \dfrac{10}{\omega\sqrt{\omega^2+1}} = 1$. Solving for ω, $\omega\sqrt{\omega^2+1} = 10$, or after squaring both sides and rearranging, $\omega^4 + \omega^2 - 100 = 0$. Solving for the roots, $\omega^2 = -10.51,\ 9.51$. Taking the square root of the positive root, we find the 0 dB frequency to be 3.08 rad/s. At this frequency, the phase angle, $\phi = -\angle(-\omega + j) = -\angle(-3.08 + j) = -162°$. Therefore the phase margin is $180° - 162° = 18°$.

b. With a delay of 0.1 s,

$$\begin{aligned}\phi &= -\angle(-\omega + j) - \omega T = -\angle(-3.08 + j) - (3.08)(0.1)(180/\text{pi}) \\ &= -162 - 17.65 = -179.65°\end{aligned}$$

Therefore the phase margin is $180° - 179.65° = 0.35°$. Thus, the system is table.

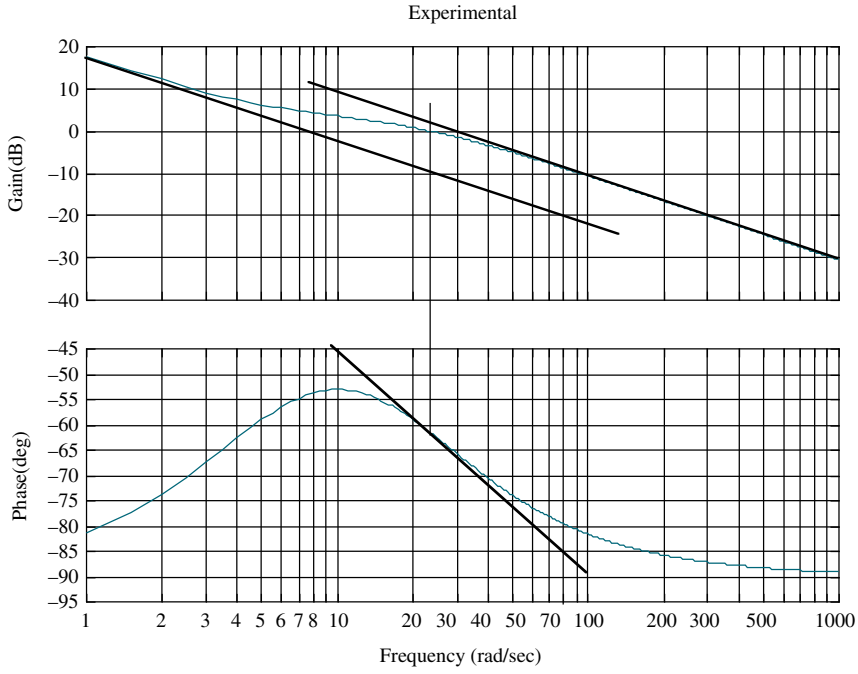
c. With a delay of 3 s,

$$\begin{aligned}\phi &= -\angle(-\omega + j) - \omega T = -\angle(-3.08 + j) - (3.08)(3)(180/\text{pi}) = -162° - 529.41° \\ &= -691.41° = 28.59 \text{ deg.}\end{aligned}$$

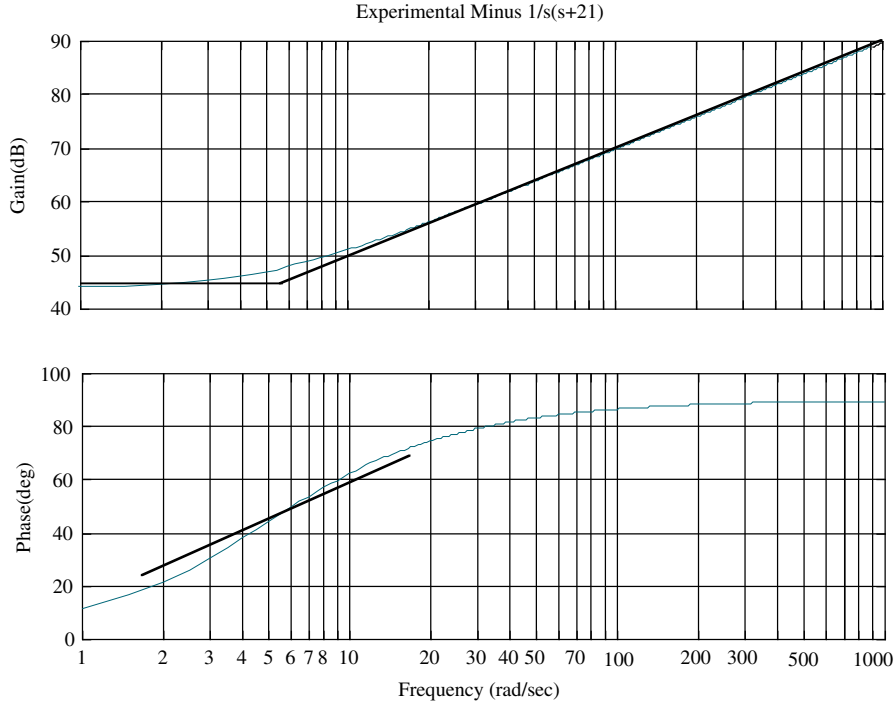
Therefore the phase margin is $28.59 - 180 = -151.41$ deg. Thus, the system is unstable.

10.12

Drawing judicially selected slopes on the magnitude and phase plot as shown below yields a first estimate.



We see an initial slope on the magnitude plot of -20 dB/dec. We also see a final -20 dB/dec slope with a break frequency around 21 rad/s. Thus, an initial estimate is $G_1(s) = \dfrac{1}{s(s+21)}$. Subtracting $G_1(s)$ from the original frequency response yields the frequency response shown below.

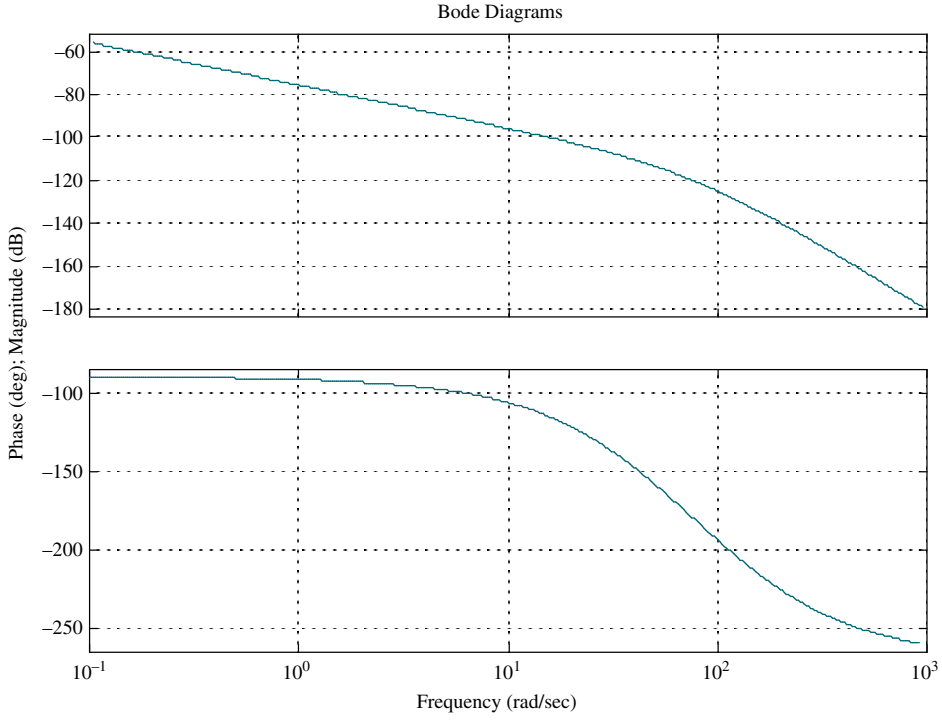


Drawing judicially selected slopes on the magnitude and phase plot as shown yields a final estimate. We see first-order zero behavior on the magnitude and phase plots with a break frequency of about 5.7 rad/s and a dc gain of about 44 dB $= 20\log(5.7K)$, or $K = 27.8$. Thus, we estimate $G_2(s) = 27.8(s + 7)$. Thus, $G(s) = G_1(s)G_2(s) = \dfrac{27.8(s + 5.7)}{s(s + 21)}$. It is interesting to note that the original problem was developed from $G(s) = \dfrac{30(s + 5)}{s(s + 20)}$.

CHAPTER 11

11.1

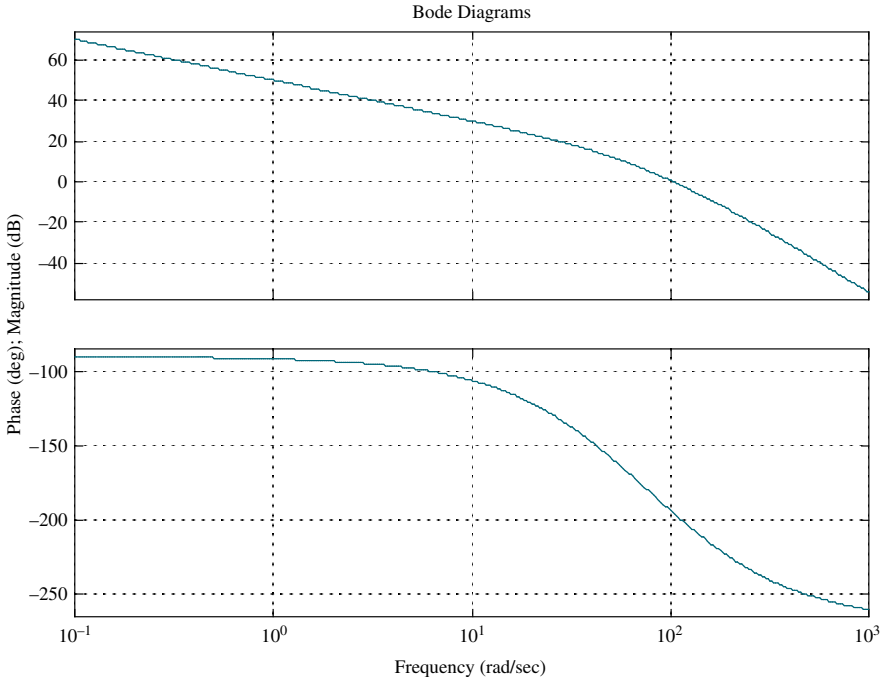
The Bode plot for $K = 1$ is shown below.



A 20% overshoot requires $\zeta = \dfrac{-\log\left(\dfrac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\dfrac{\%}{100}\right)}} = 0.456$. This damping ratio implies a phase margin of 48.10, which is obtained when the phase angle $= -180^0 + 48.10$ $= 131.9^\circ$. This phase angle occurs at $\omega = 27.6$ rad/s. The magnitude at this frequency is 5.15×10^{-6}. Since the magnitude must be unity $K = \dfrac{1}{5.15 \times 10^{-6}} = 194,200$.

11.2

To meet the steady-state error requirement, $K = 1,942,000$. The Bode plot for this gain is shown below.

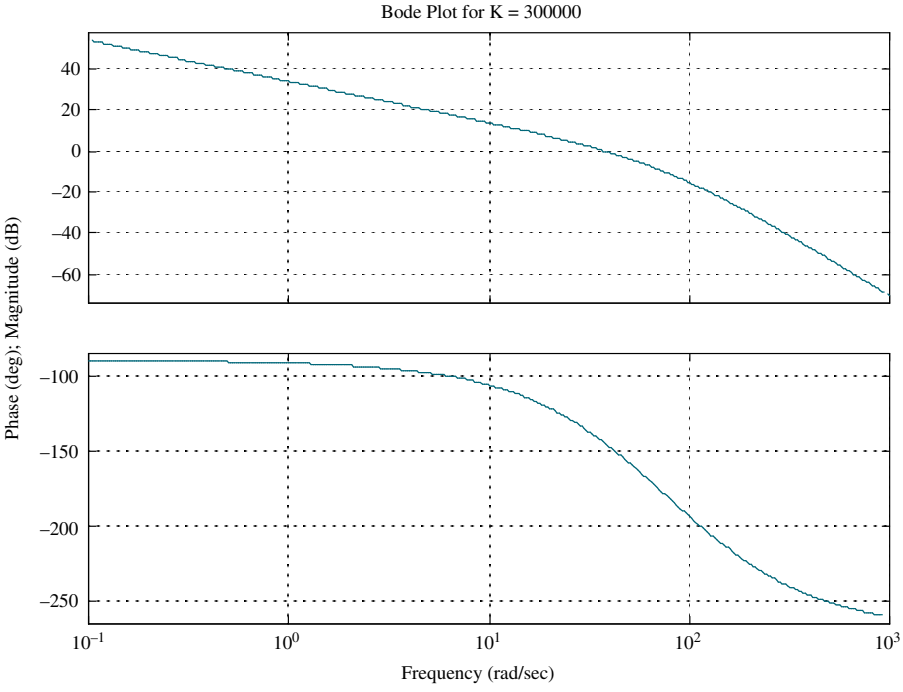


A 20% overshoot requires $\zeta = \dfrac{-\log\left(\dfrac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\dfrac{\%}{100}\right)}} = 0.456$. This damping ratio implies a phase margin of 48.1°. Adding 10° to compensate for the phase angle contribution of the lag, we use 58.1°. Thus, we look for a phase angle of $-180° + 58.1° = -129.9°$. The frequency at which this phase occurs is 20.4 rad/s. At this frequency the magnitude plot must go through zero dB. Presently, the magnitude plot is 23.2 dB. Therefore draw the high frequency asymptote of the lag compensator at −23.2 dB. Insert a break at $0.1(20.4) = 2.04$ rad/s. At this frequency, draw −23.2 dB/dec slope until it intersects 0 dB. The frequency of intersection will be the low frequency break or 0.141 rad/s. Hence the compensator is $G_c(s) = K_c\dfrac{(s+2.04)}{(s+0.141)}$, where the gain is chosen to yield 0 dB at low frequencies, or $K_c = 0.141/2.04 = 0.0691$. In summary,

$$G_c(s) = 0.0691\ \frac{(s+2.04)}{(s+0.141)} \text{ and } G(s) = \frac{1{,}942{,}000}{s(s+50)(s+120)}$$

11.3

A 20% overshoot requires $\zeta = \dfrac{-\log\left(\dfrac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\dfrac{\%}{100}\right)}} = 0.456$. The required bandwidth is then calculated as $\omega_{BW} = \dfrac{4}{T_s\zeta}\sqrt{(1-2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} = 57.9$ rad/s. In order to meet the steady-state error requirement of $K_v = 50 = \dfrac{K}{(50)(120)}$, we calculate $K = 300,000$. The uncompensated Bode plot for this gain is shown below.



The uncompensated system's phase margin measurement is taken where the magnitude plot crosses 0 dB. We find that when the magnitude plot crosses 0 dB, the phase angle is -144.8°. Therefore, the uncompensated system's phase margin is $-180^\circ + 144.8^\circ = 35.2^\circ$. The required phase margin based on the required damping ratio is $\Phi_M = \tan^{-1}\dfrac{2\zeta}{\sqrt{-2\zeta^2 + \sqrt{1+4\zeta^4}}} = 48.1^\circ$. Adding a 10° correction factor, the required phase margin is 58.1°. Hence, the compensator must contribute $\phi_{\max} = 58.1^\circ - 35.2^\circ = 22.9^\circ$. Using $\phi_{\max} = \sin^{-1}\dfrac{1-\beta}{1+\beta}$, $\beta = \dfrac{1-\sin\phi_{\max}}{1+\sin\phi_{\max}} = 0.44$. The compensator's peak magnitude is calculated as $M_{\max} = \dfrac{1}{\sqrt{\beta}} = 1.51$. Now find the frequency at which the uncompensated system has a magnitude $1/M_{\max}$, or -3.58 dB. From the Bode plot, this magnitude occurs at $\omega_{\max} = 50\,\text{rad/s}$. The compensator's zero is at $z_c = \dfrac{1}{T}$. $\omega_{\max} = \dfrac{1}{T\sqrt{\beta}}$ Therefore, $z_c = 33.2$.

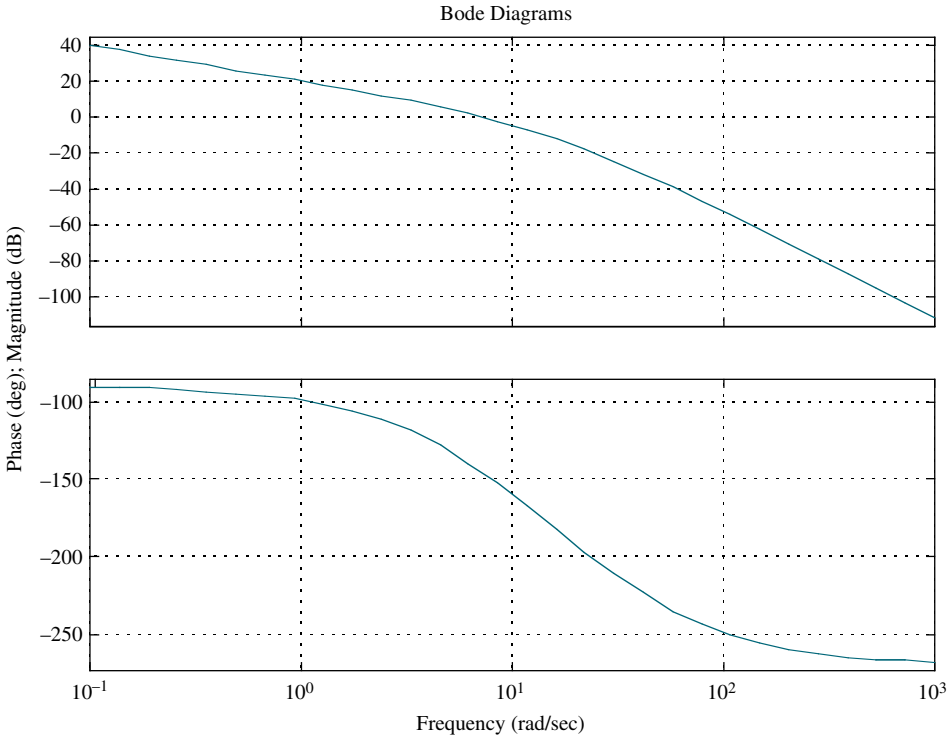
The compensator's pole is at $P_c = \dfrac{1}{\beta T} = \dfrac{z_c}{\beta} = 75.4$. The compensator gain is chosen to yield unity gain at dc.

Hence, $K_c = 75.4/33.2 = 2.27$. Summarizing, $G_c(s) = 2.27\dfrac{(s+33.2)}{(s+75.4)}$, and $G(s)\dfrac{300{,}000}{s(s+50)(s+120)}$.

11.4

A 10% overshoot requires $\zeta = \dfrac{-\log\left(\dfrac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\dfrac{\%}{100}\right)}} = 0.591$. The required bandwidth is then calculated as $\omega_{BW} = \dfrac{\pi}{T_p\sqrt{1-\zeta^2}}\sqrt{(1-2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} = 7.53\,\text{rad/}s$.

In order to meet the steady-state error requirement of $K_v = 10 = \dfrac{K}{(8)(30)}$, we calculate $K = 2400$. The uncompensated Bode plot for this gain is shown below.



Let us select a new phase-margin frequency at $0.8\omega_{BW} = 6.02\,\text{rad/s}$. The required phase margin based on the required damping ratio is $\Phi_M = \tan^{-1}\dfrac{2\zeta}{\sqrt{-2\zeta^2 + \sqrt{1+4\zeta^4}}} = 58.6^\circ$. Adding a 5° correction factor, the required phase margin is 63.6°. At 6.02 rad/s, the new phase-margin frequency, the phase angle is–which represents a phase margin of $180^\circ - 138.3^\circ = 41.7^\circ$. Thus, the lead compensator must contribute $\phi_{\max} = 63.6^\circ - 41.7^\circ = 21.9^\circ$.

Using $\phi_{\max} = \sin^{-1}\dfrac{1-\beta}{1+\beta}$, $\beta = \dfrac{1-\sin\phi_{\max}}{1+\sin\phi_{\max}} = 0.456$.

We now design the lag compensator by first choosing its higher break frequency one decade below the new phase-margin frequency, that is, $z_{lag} = 0.602\,\text{rad/s}$. The lag compensator's pole is $p_{lag} = \beta z_{lag} = 0.275$. Finally, the lag compensator's gain is $K_{lag} = \beta = 0.456$.

Now we design the lead compensator. The lead zero is the product of the new phase margin frequency and $\sqrt{\beta}$, or $z_{lead} = 0.8\omega_{BW}\sqrt{\beta} = 4.07$. Also, $p_{lead} = \frac{z_{lead}}{\beta}$ $= 8.93$. Finally, $K_{lead} = \frac{1}{\beta} = 2.19$. Summarizing,

$$G_{lag} = (s) = 0.456\frac{(s+0.602)}{(s+0.275)}; G_{lead}(s) = 2.19\frac{(s+4.07)}{(s+8.93)}; \quad \text{and } k = 2400.$$

CHAPTER 12

12.1

We first find the desired characteristic equation. A 5% overshoot requires $\zeta = \frac{-\log\left(\frac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\frac{\%}{100}\right)}} = 0.69$. Also, $\omega_n = \frac{\pi}{T_p\sqrt{1-\zeta^2}} = 14.47$ rad/s. Thus, the characteristic equation is $s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 19.97s + 209.4$. Adding a pole at -10 to cancel the zero at -10 yields the desired characteristic equation, $(s^2 + 19.97s + 209.4)(s + 10) = s^3 + 29.97s^2 + 409.1s + 2094$. The compensated system matrix in phase-variable form is

$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(k_1) & -(36+k_2) & -(15+k_3) \end{bmatrix}$. The characteristic equation for this system is $|s\mathbf{I} - (\mathbf{A} - \mathbf{BK})| = s^3 + (15+k_3)s^2 + (36+k_2)s + (k_1)$. Equating coefficients of this equation with the coefficients of the desired characteristic equation yields the gains as

$$\mathbf{K} = [k_1 \quad k_2 \quad k_3] = [2094 \quad 373.1 \quad 14.97].$$

12.2

The controllability matrix is $\mathbf{C_M} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B}] = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 4 & -9 \\ 1 & -1 & 16 \end{bmatrix}$. Since $|\mathbf{C_M}| = 80$, $\mathbf{C_M}$ is full rank, that is, rank 3. We conclude that the system is controllable.

12.3

First check controllability. The controllability matrix is $\mathrm{C}_{\mathbf{M}z} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B}] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -17 \\ 1 & -9 & 81 \end{bmatrix}$. Since $|\mathbf{C_{M}}_z| = -1$, $\mathbf{C_{Mz}}$ is full rank, that is, rank 3. We conclude that the system is controllable. We now find the desired characteristic equation. A 20% overshoot requires $\zeta = \frac{-\log\left(\frac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\frac{\%}{100}\right)}} = 0.456$. Also, $\omega_n = \frac{4}{\zeta T_s} = 4.386$ rad/s.

Thus, the characteristic equation is $s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 4s + 19.24$. Adding a pole at -6 to cancel the zero at -6 yields the resulting desired characteristic equation,

$$\left(s^2 + 4s + 19.24\right)(s + 6) = s^3 + 10s^2 + 43.24s + 115.45.$$

Since $G(s) = \dfrac{(s+6)}{(s+7)(s+8)(s+9)} = \dfrac{s+6}{s^3 + 24s^2 + 191s + 504}$, we can write the phase-variable representation as $\mathbf{A_p} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -504 & -191 & -24 \end{bmatrix}$; $\mathbf{B_p} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$; $\mathbf{C_p} = [6 \quad 1 \quad 0]$. The compensated system matrix in phase-variable form is $\mathbf{A_p} - \mathbf{B_p K_p} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(504 + k_1) & -(191 + k_2) & -(24 + k_3) \end{bmatrix}$. The characteristic equation for this system is $|s\mathbf{I} - (\mathbf{A_p} - \mathbf{B_p K_p})| = s^3 + (24 + k_3)s^2 + (191 + k_2)s + (504 + k_1)$. Equating coefficients of this equation with the coefficients of the desired characteristic equation yields the gains as $\mathbf{K_p} = [k_1 \quad k_2 \quad k_3] = [-388.55 \quad -147.76 \quad -14]$. We now develop the transformation matrix to transform back to the z-system.

$$\mathbf{C_{Mz}} = \begin{bmatrix} \mathbf{B}_z & \mathbf{A}_z\mathbf{B}_z & \mathbf{A}_z^2\mathbf{B}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -17 \\ 1 & -9 & 81 \end{bmatrix} \text{ and}$$

$$\mathbf{C_{Mp}} = \begin{bmatrix} \mathbf{B_p} & \mathbf{A_p}\mathbf{B_p} & \mathbf{A_p^2}\mathbf{B_p} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -24 \\ 1 & -24 & 385 \end{bmatrix}.$$

Therefore,

$$\mathbf{P} = \mathbf{C_{Mz}}\mathbf{C_{Mx}^{-1}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -17 \\ 1 & -9 & 81 \end{bmatrix} \begin{bmatrix} 191 & 24 & 1 \\ 24 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 7 & 1 & 0 \\ 56 & 15 & 1 \end{bmatrix}$$

Hence, $\mathbf{K}_z = \mathbf{K_p}\mathbf{P}^{-1} = [-388.55 \quad -147.76 \quad -14] \begin{bmatrix} 1 & 0 & 0 \\ -7 & 1 & 0 \\ 49 & -15 & 1 \end{bmatrix}$

$= [-40.23 \quad 62.24 \quad -14]$.

12.4

For the given system $\mathbf{e}_{\dot{\mathbf{x}}} = (\mathbf{A} - \mathbf{LC})\mathbf{e}_{\mathbf{x}} = \begin{bmatrix} -(24 + l_1) & 1 & 0 \\ -(191 + l_2) & 0 & 1 \\ -(504 + l_3) & 0 & 0 \end{bmatrix} \mathbf{e}_{\mathbf{x}}$. The characteristic polynomial is given by $|[s\mathbf{I} - (\mathbf{A} - \mathbf{LC})]| = s^3 + (24 + l_1)s^2 + (191 + l_2)s + (504 + l_3)$. Now we find the desired characteristic equation. The dominant poles from Skill-Assessment Exercise 12.3 come from $\left(s^2 + 4s + 19.24\right)$. Factoring yields $(-2 + j3.9)$ and $(-2 - j3.9)$. Increasing these poles by a factor of 10 and adding a third pole 10 times the real part of the dominant second-order poles yields the

desired characteristic polynomial, $(s+20+j39)(s+20-j39)(s+200) = s^3 + 240s^2 + 9921s + 384200$. Equating coefficients of the desired characteristic equation to the system's characteristic equation yields $\mathbf{L} = \begin{bmatrix} 216 \\ 9730 \\ 383696 \end{bmatrix}$.

12.5

The observability matrix is $\mathbf{O_M} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} = \begin{bmatrix} 4 & 6 & 8 \\ -64 & -80 & -78 \\ 674 & 848 & 814 \end{bmatrix}$, where $\mathbf{A}^2 = \begin{bmatrix} 25 & 28 & 32 \\ -7 & -4 & -11 \\ 77 & 95 & 94 \end{bmatrix}$. The matrix is of full rank, that is, rank 3, since $|\mathbf{O_M}| = -1576$. Therefore the system is observable.

12.6

The system is represented in cascade form by the following state and output equations:

$$\dot{\mathbf{z}} = \begin{bmatrix} -7 & 1 & 0 \\ 0 & -8 & 1 \\ 0 & 0 & -9 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [1 \quad 0 \quad 0]\mathbf{z}$$

The observability matrix is $\mathbf{O_{Mz}} = \begin{bmatrix} \mathbf{C_z} \\ \mathbf{C_z A_z} \\ \mathbf{C_z A_z^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -7 & 1 & 0 \\ 49 & -15 & 1 \end{bmatrix}$,

where $\mathbf{A_z^2} = \begin{bmatrix} 49 & -15 & 1 \\ 0 & 64 & -17 \\ 0 & 0 & 81 \end{bmatrix}$. Since $G(s) = \dfrac{1}{(s+7)(s+8)(s+9)}$

$= \dfrac{1}{s^3 + 24s^2 + 191s + 504}$, we can write the observable canonical form as

$$\dot{\mathbf{x}} = \begin{bmatrix} -24 & 1 & 0 \\ -191 & 0 & 1 \\ -504 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [1 \quad 0 \quad 0]\mathbf{x}$$

The observability matrix for this form is $\mathbf{O_{Mx}} = \begin{bmatrix} \mathbf{C_x} \\ \mathbf{C_x A_x} \\ \mathbf{C_x A_x^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -24 & 1 & 0 \\ 385 & -24 & 1 \end{bmatrix}$,

where

$$\mathbf{A}_\mathbf{x}^2 = \begin{bmatrix} 385 & -24 & 1 \\ 4080 & -191 & 0 \\ 12096 & 504 & 0 \end{bmatrix}.$$

We next find the desired characteristic equation. A 10% overshoot requires $\zeta = \dfrac{-\log\left(\dfrac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\dfrac{\%}{100}\right)}} = 0.591$. Also, $\omega_n = \dfrac{4}{\zeta T_s} = 67.66$ rad/s. Thus, the characteristic equation is $s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 80s + 4578.42$. Adding a pole at -400, or 10 times the real part of the dominant second-order poles, yields the resulting desired characteristic equation, $\left(s^2 + 80s + 4578.42\right)(s + 400) = s^3 + 480s^2 + 36580s + 1.831x10^6$. For the system represented in observable canonical form $\mathbf{e}_{\dot{\mathbf{x}}} = (\mathbf{A}_\mathbf{x} - \mathbf{L}_\mathbf{x}\mathbf{C}_\mathbf{x})\,\mathbf{e}_\mathbf{x} = \begin{bmatrix} -(24 + l_1) & 1 & 0 \\ -(191 + l_2) & 0 & 1 \\ -(504 + l_3) & 0 & 0 \end{bmatrix}\mathbf{e}_\mathbf{x}$. The characteristic polynomial is given by $|[s\mathbf{I} - (\mathbf{A}_\mathbf{x} - \mathbf{L}_\mathbf{x}\mathbf{C}_\mathbf{x})]| = s^3 + (24 + l_1)s^2 + (191 + l_2)s + (504 + l_3)$. Equating coefficients of the desired characteristic equation to the system's characteristic equation yields $\mathbf{L}_\mathbf{x} = \begin{bmatrix} 456 \\ 36,389 \\ 1,830,496 \end{bmatrix}$.

Now, develop the transformation matrix between the observer canonical and cascade forms.

$$\begin{aligned} \mathbf{P} = \mathbf{O}_{\mathbf{Mz}}^{-1}\mathbf{O}_{\mathbf{Mx}} &= \begin{bmatrix} 1 & 0 & 0 \\ -7 & 1 & 0 \\ 49 & -15 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ -24 & 1 & 0 \\ 385 & -24 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 7 & 1 & 0 \\ 56 & 15 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -24 & 1 & 0 \\ 385 & -24 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ -17 & 1 & 0 \\ 81 & -9 & 1 \end{bmatrix} \end{aligned}$$

Finally,

$$\mathbf{L}_\mathbf{z} = \mathbf{P}\mathbf{L}_\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \\ -17 & 1 & 0 \\ 81 & -9 & 1 \end{bmatrix} \begin{bmatrix} 456 \\ 36,389 \\ 1,830,496 \end{bmatrix} = \begin{bmatrix} 456 \\ 28,637 \\ 1,539,931 \end{bmatrix} \approx \begin{bmatrix} 456 \\ 28,640 \\ 1,540,000 \end{bmatrix}.$$

12.7

We first find the desired characteristic equation. A 10% overshoot requires

$$\zeta = \frac{-\log\left(\frac{\%}{100}\right)}{\sqrt{\pi^2 + \log^2\left(\frac{\%}{100}\right)}} = 0.591$$

Also, $\omega_n = \dfrac{\pi}{T_p\sqrt{1-\zeta^2}} = 1.948\,\text{rad/s}$. Thus, the characteristic equation is $s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 2.3s + 3.79$. Adding a pole at -4, which corresponds to the original system's zero location, yields the resulting desired characteristic equation, $(s^2 + 2.3s + 3.79)(s+4) = s^3 + 6.3s^2 + 13s + 15.16$.

Now, $\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{BK}) & \mathbf{B}K_e \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_N \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r$; and $y = [\mathbf{C} \quad 0] \begin{bmatrix} \mathbf{x} \\ x_N \end{bmatrix}$,

where

$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 0 & 1 \\ -7 & -9 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} [k_1 \quad k_2] = \begin{bmatrix} 0 & 1 \\ -7 & -9 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ -(7+k_1) & -(9+k_2) \end{bmatrix}$$

$$\mathbf{C} = [4 \quad 1]$$

$$\mathbf{B}k_e = \begin{bmatrix} 0 \\ 1 \end{bmatrix} k_e = \begin{bmatrix} 0 \\ k_e \end{bmatrix}$$

Thus,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -(7+k_1) & -(9+k_2) & k_e \\ -4 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_N \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r;\; y = [4 \quad 1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_N \end{bmatrix}.$$

Finding the characteristic equation of this system yields

$$\left| s\mathbf{I} - \begin{bmatrix} (\mathbf{A} - \mathbf{BK}) & \mathbf{B}K_e \\ -\mathbf{C} & 0 \end{bmatrix} \right| = \left| \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ -(7+k_1) & -(9+k_2) & k_e \\ -4 & -1 & 0 \end{bmatrix} \right|$$

$$= \left| \begin{bmatrix} s & -1 & 0 \\ (7+k_1) & s+(9+k_2) & -k_e \\ 4 & 1 & s \end{bmatrix} \right| = s^3 + (9+k_2)s^2 + (7+k_1+k_e)s + 4k_e$$

Equating this polynomial to the desired characteristic equation,

$$s^3 + 6.3s^2 + 13s + 15.16 = s^3 + (9+k_2)s^2 + (7+k_1+k_e)s + 4k_e$$

Solving for the k's,

$$\mathbf{K} = [2.21 \quad -2.7] \text{ and } k_e = 3.79.$$

CHAPTER 13

13.1

$$f(t) = \sin(\omega kT);\ f^*(t) = \sum_{k=0}^{\infty} \sin(\omega kT)\delta(t - kT);$$

$$F^*(s) = \sum_{k=0}^{\infty} \sin(\omega kT)e^{-kTs} = \sum_{k=0}^{\infty} \frac{\left(e^{j\omega kT} - e^{-j\omega kT}\right)e^{-kTs}}{2j}$$

$$= \frac{1}{2j}\sum_{k=0}^{\infty}\left(e^{T(s-j\omega)}\right)^{-k} - \left(e^{T(s+j\omega)}\right)^{-k}$$

But, $\sum_{k=0}^{\infty} x^{-k} = \frac{1}{1 - x^{-1}}$

Thus,

$$F^*(s) = \frac{1}{2j}\left[\frac{1}{1 - e^{-T(s-j\omega)}} - \frac{1}{1 - e^{-T(s+j\omega)}}\right] = \frac{1}{2j}\left[\frac{e^{-Ts}e^{j\omega T} - e^{-Ts}e^{j\omega T}}{1 - (e^{-Ts}e^{j\omega T} - e^{-Ts}e^{j\omega T}) + e^{-2Ts}}\right]$$

$$= e^{-Ts}\left[\frac{\sin(\omega T)}{1 - e^{-Ts}2\cos(\omega T) + e^{-2Ts}}\right] = \frac{z^{-1}\sin(\omega T)}{1 - 2z^{-1}\cos(\omega T) + z^{-2}}$$

13.2

$$F(z) = \frac{z(z+1)(z+2)}{(z-0.5)(z-0.7)(z-0.9)}$$

$$\frac{F(z)}{z} = \frac{z(z+1)(z+2)}{(z-0.5)(z-0.7)(z-0.9)}$$

$$= 46.875\frac{z}{z-0.5} - 114.75\frac{1}{z-0.7} + 68.875\frac{z}{z-0.9}$$

$$F(z) = 46.875\frac{z}{z-0.5} - 114.75\frac{z}{z-0.7} + 68.875\frac{z}{z-0.9},$$

$$f(kT) = 46.875(0.5)^k - 114.75(0.7)^k + 68.875(0.9)^k$$

13.3

Since $G(s) = \left(1 - e^{-Ts}\right)\frac{8}{s(s+4)}$,

$$G(z) = \left(1 - z^{-1}\right)z\left\{\frac{8}{s(s+4)}\right\} = \frac{z-1}{z}z\left\{\frac{A}{s} + \frac{B}{s+4}\right\} = \frac{z-1}{z}z\left\{\frac{2}{s} + \frac{2}{s+4}\right\}.$$

Let $G_2(s) = \frac{2}{s} + \frac{2}{s+4}$. Therefore, $g_2(t) = 2 - 2e^{-4t}$, or $g_2(kT) = 2 - 2e^{-4kT}$.

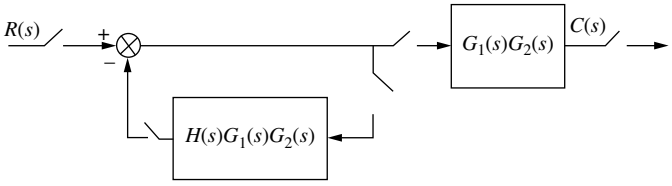
Hence, $G_2(z) = \frac{2z}{z-1} - \frac{2z}{z - \mathrm{e}^{-4T}} = \frac{2z\left(1 - \mathrm{e}^{-4T}\right)}{(z-1)(z - e^{-4T})}$.

Therefore, $G(z) = \dfrac{z-1}{z} G_2(z) = \dfrac{2\left(1-e^{-4T}\right)}{\left(z-e^{-4T}\right)}$.

For $T = \dfrac{1}{4}s$, $G(z) = \dfrac{1.264}{z-0.3679}$.

13.4

Add phantom samplers to the input, feedback after $H(s)$, and to the output. Push $G_1(s)G_2(s)$, along with its input sampler, to the right past the pickoff point and obtain the block diagram shown below.



Hence, $T(z) = \dfrac{G_1G_2(z)}{1+HG_1G_2(z)}$.

13.5

Let $G(s) = \dfrac{20}{s+5}$. Let $G_2(s) = \dfrac{G(s)}{s} = \dfrac{20}{s(s+5)} = \dfrac{4}{s} - \dfrac{4}{s+5}$. Taking the inverse Laplace transform and letting $t = kT$, $g_2(kT) = 4 - 4e^{-5kT}$. Taking the z-transform yields $G_2(z) = \dfrac{4z}{z-1} - \dfrac{4z}{z-e^{-5T}} = \dfrac{4z\left(1-e^{-5T}\right)}{(z-1)(z-e^{-5T})}$.

Now, $G(z) = \dfrac{z-1}{z} - G_2(z) = \dfrac{4\left(1-e^{-5T}\right)}{\left(z-e^{-5T}\right)}$.

Finally, $T(z) = \dfrac{G(z)}{1+G(z)} = \dfrac{4\left(1-e^{-5T}\right)}{z-5e^{-5T}+4}$.

The pole of the closed-loop system is at $5e^{-5T} - 4$. Substituting values of T, we find that the pole is greater than 1 if $T > 0.1022$ s. Hence, the system is stable for $0 < T < 0.1022$ s.

13.6

Substituting $z = \dfrac{s+1}{s-1}$ into $D(z) = z^3 - z^2 - 0.5z + 0.3$, we obtain $D(s) = s^3 - 8s^2 - 27s - 6$. The Routh table for this polynomial is shown below.

s^3	1	−27
s^2	−8	−6
s^1	−27.75	0
s^0	−6	0

Since there is one sign change, we conclude that the system has one pole outside the unit circle and two poles inside the unit circle. The table did not produce a row of zeros and thus, there are no $j\omega$ poles. The system is unstable because of the pole outside the unit circle.

13.7

Defining $G(s)$ as $G_1(s)$ in cascade with a zero-order-hold,

$$G(s) = 20\big(1 - e^{-Ts}\big)\left[\frac{(s+3)}{s(s+4)(s+5)}\right] = 20\big(1 - e^{-Ts}\big)\left[\frac{3/20}{s} + \frac{1/4}{(s+4)} - \frac{2/5}{(s+5)}\right].$$

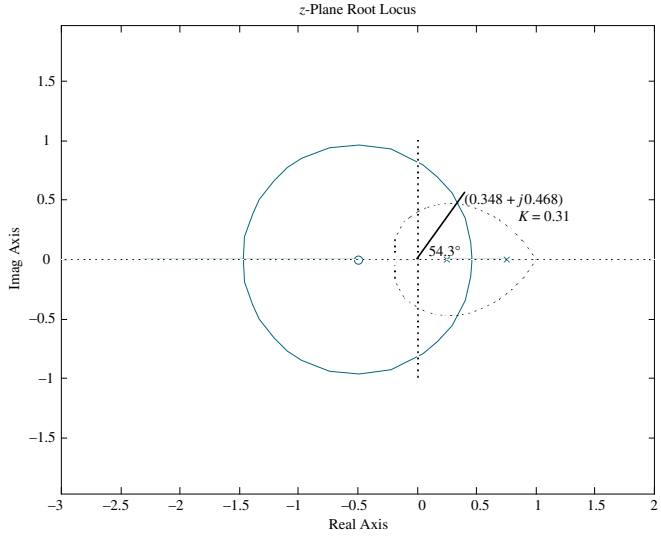
Taking the z-transform yields

$$G(z) = 20\big(1 - z^{-1}\big)\left[\frac{(3/20)z}{z-1} + \frac{(1/4)z}{z - e^{-4T}} - \frac{(2/5)z}{z - e^{-5T}}\right] = 3 + \frac{5(z-1)}{z - e^{-4T}} - \frac{8(z-1)}{z - e^{-5T}}.$$

Hence for $T = 0.1$ second, $K_p = \lim_{z\to 1} G(z) = 3$, and $K_v = \frac{1}{T}\lim_{z\to 1}(z-1)G(z) = 0$, and $K_a = \frac{1}{T^2}\lim_{z\to 1}(z-1)^2 G(z) = 0$. Checking for stability, we find that the system is stable for $T = 0.1$ second, since $T(z) = \frac{G(z)}{1+G(z)} = \frac{1.5z - 1.109}{z^2 + 0.222z - 0.703}$ has poles inside the unit circle at -0.957 and $+0.735$. Again, checking for stability, we find that the system is unstable for $T = 0.5$ second, since $T(z) = \frac{G(z)}{1+G(z)} = \frac{3.02z - 0.6383}{z^2 + 2.802z - 0.6272}$ has poles inside and outside the unit circle at $+0.208$ and -3.01, respectively.

13.8

Draw the root locus superimposed over the $\zeta = 0.5$ curve shown below. Searching along a 54.3° line, which intersects the root locus and the $\zeta = 0.5$ curve, we find the point $0.587\angle 54.3^\circ = (0.348 + j0.468)$ and $K = 0.31$.

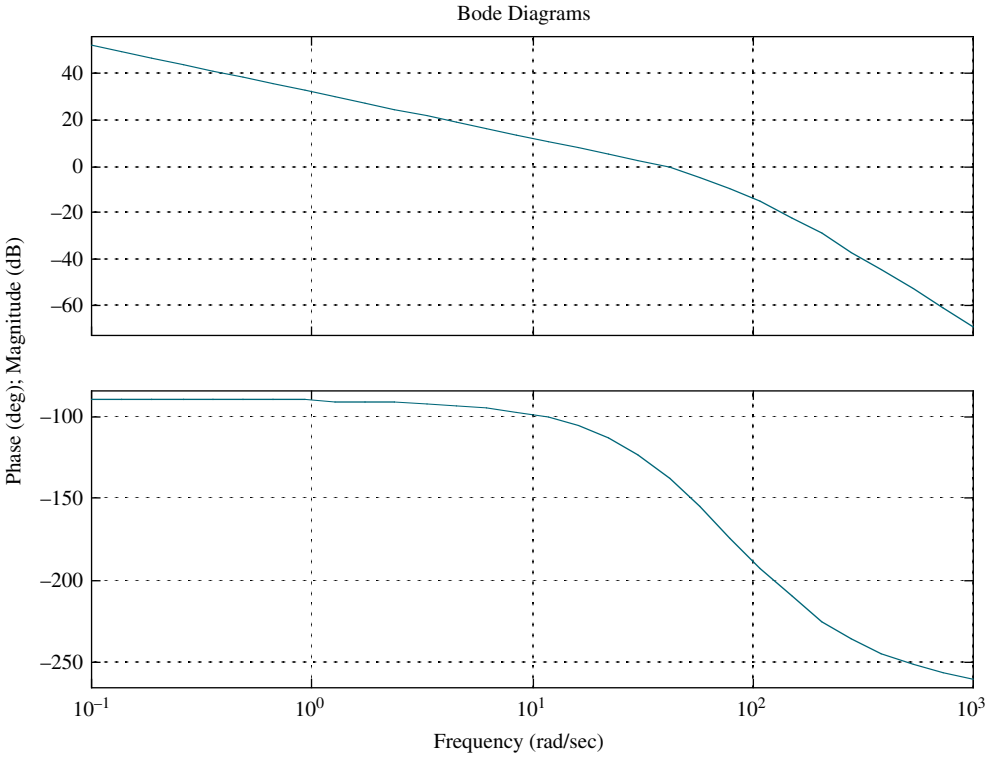


13.9

Let

$$G_e(s) = G(s)G_c(s) = \frac{100K}{s(s+36)(s+100)}\frac{2.38(s+25.3)}{(s+60.2)} = \frac{342720(s+25.3)}{s(s+36)(s+100)(s+60.2)}.$$

The following shows the frequency response of $G_e(j\omega)$.



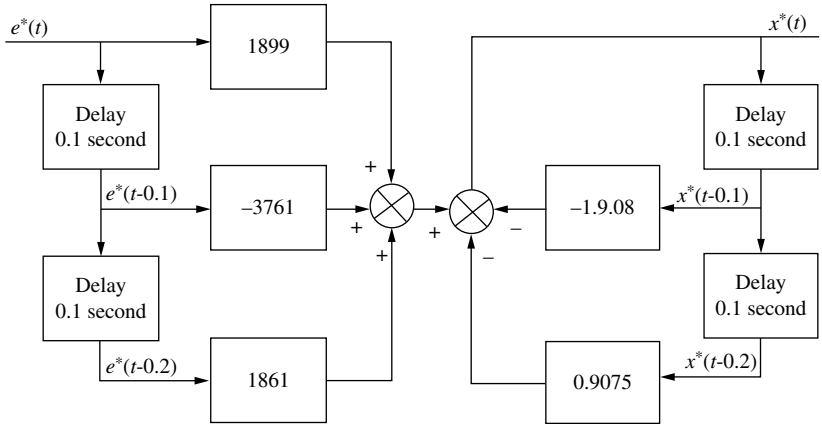
We find that the zero dB frequency, ω_{Φ_M}, for $G_e(j\omega)$ is 39 rad/s. Using Astrom's guideline the value of T should be in the range, $0.15/\omega_{\Phi_M} = 0.0038$ second to $0.5/\omega_{\Phi_M} = 0.0128$ second. Let us use $T = 0.001$ second. Now find the Tustin transformation for the compensator. Substituting $s = \dfrac{2(z-1)}{T(z-1)}$ into $G_c(s) = \dfrac{2.38(s+25.3)}{(s+60.2)}$ with $T = 0.001$ second yields

$$G_c(z) = 2.34\frac{(z-0.975)}{(z-0.9416)}.$$

13.10

$G_c(z) = \dfrac{X(z)}{E(z)} = \dfrac{1899z^2 - 3761z + 1861}{z^2 - 1.908z + 0.9075}$. Cross-multiply and obtain $\left(z^2 - 1.908z + 0.9075\right)X(z) = \left(1899z^2 - 3761z + 1861\right)E(z)$. Solve for the highest power of z operating on the output, $X(z)$, and obtain $z^2X(z) = \left(1899z^2 - 3761z + 1861\right)E(z) - (-1.908z + 0.9075)X(z)$. Solving for $X(z)$ on the left-hand side yields

$X(z) = \left(1899 - 3761z^{-1} + 1861z^{-2}\right)\ E(z) - \left(-1.908z^{-1} + 0.9075z^{-2}\right)X(z)$. Finally, we implement this last equation with the following flow chart:



Copyright © 2011 by John Wiley & Sons, Inc.

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work. In 2008, we launched a Corporate Citizenship Initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: www.wiley.com/go/citizenship.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774, (201) 748-6011, fax (201) 748-6008 or online at http://www.wiley.com/go/permissions.

Evaluation copies are provided to qualified academics and professionals for review purposes only, for use in their courses during the next academic year. These copies are licensed and may not be sold or transferred to a third party. Upon completion of the review period, please return the evaluation copy to Wiley. Return instructions and a free of charge return shipping label are available at www.wiley.com/go/returnlabel. Outside of the United States, please contact your local representative.

ISBN 13 978-0470-54756-4

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1